Multi-Armed Bandits with Correlated Arms

EE 6106 Project

Aziz Shameem
Dept. of Electrical Engineering
IIT Bombay
20d070020@iitb.ac.in

Sameep Chattopadhyay
Dept. of Electrical Engineering
IIT Bombay
20d070067@iitb.ac.in

I. Background

In this study, we have explored the domain of multi-armed bandits (MAB), specifically focusing on the scenario of correlated arms. The majority of the classical MAB algorithms implicitly assume that the rewards generated by the arms are unconnected to one another and that selecting one arm does not convey any additional insights into the remaining K-1 arms. Nevertheless, in numerous practical applications, such as those involving treatments, drugs, or advertisements, it is probable that the arms are correlated with one another, thereby invalidating this assumption. [1] has developed an approach to leverage these reward correlations and generalize the classic bandit algorithms according to the correlated setting; it also contains a rigorous analysis for the case of the UCB algorithm, with and without correlation showing that the presence of information about the correlation between arms provides an opportunity for higher performance as compared to applying the algorithm naively to the independent arms.

II. Problem Statement

The MAB problem falls under the class of sequential decision-making problems. The classical multi-armed bandit problem has a set of K arms, with each arm having an unknown reward distribution. At the beginning of each round, t, we need to decide an arm $k_t \in$ K, and based on our decision, we receive a random reward R drawn from the reward distribution of arm kt. The classical multi-armed bandit aims to maximize the long-term cumulative reward. We here study the case of correlated arms where the reward distribution satisfies the criteria that $E[R_l|R_k] \neq E[R_l]$ for some l,k $\leq$K.

One way of capturing these correlations is through the knowledge of the joint reward distribution. However, if the complete joint reward distribution is known, then the best arm is known trivially. Therefore we apply a more non-restrictive form of correlation by assuming knowledge of an upper bound on the conditional expected rewards, which the paper has defined as a term called Pseudo-Rewards, denoted by $s_{l,k}(r)$ where-

$$E[R_l|R_k = r] \leq s_{l,k}(r)$$

The findings indicate that the possession of such bounds, even if they are not tight, has the potential to result in significant enhancements in the cumulative reward obtained by reducing the extent of exploration in contrast to classical Multi-Armed Bandit (MAB) algorithms. The proposed MAB framework and algorithm can be employed across all real-world applications of the classical MAB algorithms in instances where it is viable to estimate pseudo-rewards through domain expertise or via surveyed data.

A. Recommender System Using Correlated Rewards

The setup described in the paper corresponds to a global recommender system using correlated rewards bandits where the recommender does not observe the user's contextual (age/occupational/income) features and hence cannot provide personalized recommendations; instead, it aims to provide global recommendations to a population whose demographics is unknown. Taking the example of a movie recommender system, a user reacting positively to the movie *Kuch Kuch Hota Hai* might also be more likely to react positively to the movie *Kabhi Khushi Kabhie Gham*, which belongs to the same genre and has been made by the same director and lead cast as the earlier film, such examples can also be produced for negative correlation.

B. Computing Pseudo-Rewards

As discussed earlier, the pseudo-rewards can be learned from prior-available data or through offline surveys in which users are presented with all K arms allowing them to sample the rewards jointly. Else in the presence of a training dataset, we can compute the pseudo-rewards using the following steps-

1) For large training data, one can estimate empirical mean for joint distribution ($\hat{\mu}_{l,k}(r)$) as the average reward for arm l for all users with reward r for arm k.
2) We can show that $\hat{\mu}_{l,k}(r)$ is an unbiased estimator for $E[R_l|R_k = r]$, and by the law of large numbers, they converge for large dataset.
3) We can now use the value $\hat{\mu}_{l,k}(r)+\Delta$ as an upper-bound on $E[R_l|R_k = r]$, here Δ is known as the safety buffer.
4) One suggested value for the Δ is $\hat{\sigma}_{l,k}(r)$; thus, we take $s_{l,k}(r) = \hat{\mu}_{l,k}(r)+\hat{\sigma}_{l,k}(r)$ while the value is lesser than the maximum possible reward.
5) Lastly, the pseudo-rewards for any unknown conditional mean reward could be filled with the maximum possible reward for the corresponding arm

Note that in the case that all pseudo-reward entries are unknown, then all pseudo-reward entries can be replaced with

the maximum possible reward for each arm. The case would then reduce to classic MAB.

C. Empirical Pseudo-Rewards

In the correlated MAB setup, the pseudo-reward of arm l w.r.t arm k is used to estimate the arm l's reward through the reward sample from arm k. The paper defines the quantity empirical pseudo-reward which can be used to obtain an optimistic estimate for μ_l using reward samples of arm k. The definition is lifted as is from [1]-

Definition (Empirical and Expected Pseudo-Reward). *"After t rounds, arm k is pulled $n_k(t)$ times. Using this $n_k(t)$ reward realizations, we can construct the empirical pseudo-reward ($\hat{\phi}_{l,k}(t)$) for each arm l w.r.t. arm k as follows-*

$$\hat{\phi}_{l,k}(t) \triangleq \frac{\Sigma_{\tau=1}^{t} \mathcal{I}_{k_\tau=k}(r_{k_\tau}) s_{l,k}(r)}{n_k(t)}$$

Note that the empirical pseudo-reward $\hat{\phi}_{l,k}(t)$ is defined with respect to arm k, and it is only a function of the rewards observed by pulling arm k." For the online framework, all the computations will be done according to the value of $\hat{\phi}_{l,k}(t)$

III. Methods

A. Epsilon-Greedy

This is the most basic algorithm in a multi-armed bandit setting. In order to maintain exploration while also limiting the regret by exploiting, the algorithm maintains empirical means of the rewards $\hat{\mu}_k(t)$. and with probability $1-\epsilon$ chooses the arm A_{t+1} = $\text{argmax}_k(\hat{\mu}_k(t))$, and samples randomly with probability ϵ. This has been shown to work better when the true mean distribution is symmetrical. Along with the original Epsilon-Greedy algorithm, we have tried two other variants-

1) **With Constant Learning Rate:** In this method, a learning rate (α) is defined and $\hat{\mu}_k(t)$ is updated as per the following equation-

$$\hat{\mu}_k(t+1) = \hat{\mu}_k(t) + \alpha(R_t - \hat{\mu}_k(t))\mathcal{I}_{A_t=k}$$

The epsilon-greedy algorithm with a constant learning rate uses a fixed learning rate for updating the expected rewards at each step. This can make the learning process more stable and easier to control, but it can also make it slower to adapt to changes in the environment.

2) **With Optimistic Initial Values:** In this method, we take $\hat{\mu}_k(0)$ equal to the maximum possible reward for arm k and $\hat{\mu}_k(t)$ is updated as per the following equation-

$$\hat{\mu}_k(t+1) = \hat{\mu}_k(t) + \frac{R_t - \hat{\mu}_k(t)}{n_k(t)+1} * \mathcal{I}_{A_t=k}$$

The epsilon-greedy algorithm with an optimistic initial value gives more weightage to the initial pulls and starts with a high expectation. This algorithm is known to perform best when most arms have high mean reward.

B. UCB

Exploration is needed because there is always uncertainty about the accuracy of the action-value estimates. The greedy actions are those that look best at present, but some of the other actions may actually be better. "-greedy action selection forces the non-greedy actions to be tried indiscriminately, with no preference for those nearly greedy or uncertain. It would be better to select among the non-greedy actions according to their potential for actually being optimal, taking into account both how close their estimates are to being maximal and the uncertainties in those estimates. One effective way of doing this is to select actions according to

$$A_t = \underset{a}{\text{argmin}} \left[Q_t(a) + \sqrt{\frac{lnt}{N_t(a)}} \right]$$

where $N_t(a)$ denotes the number of times that action a has been selected prior to time t, and the number $c > 0$ controls the degree of exploration. $Q_t(a)$ can correspond to the empirical means of the observations upto time t.

C. Thompson Sampling

Thompson Sampling (Posterior Sampling or Probability Matching) is an algorithm for choosing the actions that address the exploration-exploitation dilemma in the multi-armed bandit problem.

Thompson Sampling takes a slightly different approach; rather than just refining an estimate of the mean reward, it extends this instead to build up a probability model from the obtained rewards, and then samples from this to choose an action. In this way, not only is an increasingly accurate estimate of the possible reward obtained, but the model also provides a level of confidence in this reward, and this confidence increases as more samples are collected. This process of updating your beliefs as more evidence becomes available is known as Bayesian Inference.

Under Thompson sampling, the arm $k_{t+1} = \text{argmax}_{k \in \mathcal{K}} S_{k,t}$ is selected at time step $t+1$. Here, $S_{k,t}$ is the sample obtained from the posterior distribution of μ_k, That is,

$$k_{t+1} = \underset{k \in \mathcal{K}}{\text{argmax}}\, S_{k,t}$$
$$S_{k,t} \sim \mathcal{N}\left(\hat{\mu_k}(t), \frac{\beta B}{n_k(t)+1}\right)$$

D. Gradient Bandit

Here, we consider learning a numerical preference for each action a, which we denote $H_t(a)$. The larger the preference, the more often that action is taken, but the preference has no interpretation in terms of reward. Only the relative preference of one action over another is important, which are determined according to a soft-max distribution (i.e., Gibbs or Boltzmann distribution) as follows:

$$\mathrm{P}(A_t = a) = \frac{e^{H_t(a)}}{\sum_{b=1}^{k} e^{H_t(b)}}$$

There is a natural learning algorithm for this setting based on the idea of stochastic gradient ascent. On each step, after selecting action At and receiving the reward R_t, the action preferences are updated by :

$$H_{t+1}(A_t) = H_t(A_t) + \alpha(R_t - \bar{R}_t)(1 - \mathrm{P}(A_t = A_t))$$
$$H_{t+1}(a) = H_t(a) - \alpha(R_t - \bar{R}_t)(\mathrm{P}(a)) \text{ for } a \neq A_t$$

where $\alpha > 0$ is a step-size parameter, and $\bar{R}_t \in R$ is the average of all the rewards up through and including time t, which can be computed incrementally.

E. C-Bandit Algorithm

Now that we have an idea about the existing bandit algorithms and the empirical mean, we now describe the procedure for utilizing the correlated MAB setting for any generalized bandit algorithm-

1) **Identify Significant Arms:** At each round t, define S_t as the set of arms that have at least t/K samples, where K is the total number of arms; S_t ={ $k \in \mathcal{K}$: $n_k(t) \geq$ t/K}. Next, define $k^m(t)$ to be the arm with the highest empirical mean in set S_t and let $\hat{\mu}_{k^m}(t)$ be the empirical mean
2) **Identify Competitive Arms:** We use the value of $\hat{\mu}_{k^m}(t)$ to define non-competitive arms; an arm k is said to be Non-Competitive at round t, if-

$$min_{l \in S_t}(\hat{\phi}_{l,k}(t)) \leq \hat{\mu}_{k^m}(t)$$

$min_{l \in S_t}(\hat{\phi}_{l,k}(t))$ provides the tightest estimated upper bound on the mean of arms k. If this estimated upper bound is smaller than $\hat{\mu}_{k^m}(t)$, then we call arm k as non-competitive as it seems unlikely to be optimal through the samples of arms in S_t. Any arm k $\notin S_t$ is considered to be an optimal arm. Note that the set of empirically competitive and empirically non-competitive arms is evaluated at each round t. Hence, an arm that is empirically non-competitive at round t may be empirically competitive in subsequent rounds.
3) **Play Algorithm on Competitive Arms:** Now play the original MAB algorithm on the set of optimal arms

IV. Experiments and Results

Dataset

The dataset referred to as MovieLens [3] encompasses a grand total of 1 million ratings, obtained from 6040 users, for a compilation of 3883 movies. The ratings are assigned by users on a 1-5 scale, while each movie is affiliated with one (and, in certain cases, more than one) genre. One genre is arbitrarily selected from the possibly several genres linked to each movie for our experimental purposes.

To conduct the experiments, the data was partitioned into two sets: the first half encompasses the ratings assigned by the users who submitted the greatest number of ratings and is utilized to learn the pseudo-reward entries. The other half, which represents the test set, is exploited to assess the efficacy of the proposed algorithms. This division guarantees that the rating allocation differs between the training and test datasets.

A. Genre Recommendation

The following plots summarise the results obtained on running the six algorithms and their corresponding *C_algos* for recommending genres.

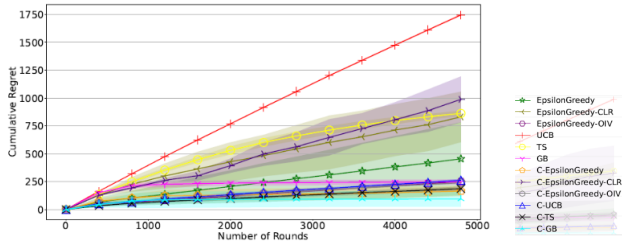

Fig. 1: Plot showing the evolution of regret with the number of pulls

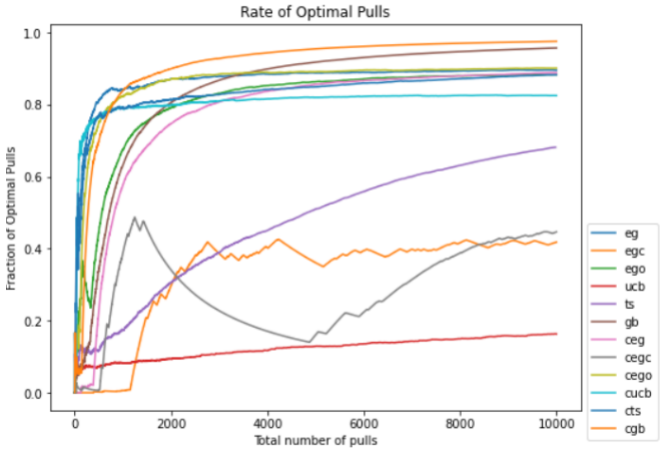

Fig. 2: The fraction of optimal pulls against the total number of arm pulls

The observations from the plots are summarised as follows :

1) For all algorithms, the correlated versions outperform the corresponding uncorrelated algorithms.
2) Gradient Bandit algorithms seem to give the best performance, both in terms of regret as well as the number of optimal pulls.
3) The instability of Epsilon Greedy algorithm with constant learning rate is apparent from 4, where decreasing lines indicate large time frames where the optimal arm was not pulled at all.
4) 4 also shows the rate at which algorithms learn. UCB, Thompson Sampling and constant-learning-rate-Epsilon Greedy seems to be slower at getting to the best arm than the rest of the algorithms.

5) The variance of the probabilistic algorithms can be seen in 3. The epsilon greedy algorithm, for instance, assumes greater variance due to its heavy reliance on probabilistic outcomes.

B. Movie Recommendation

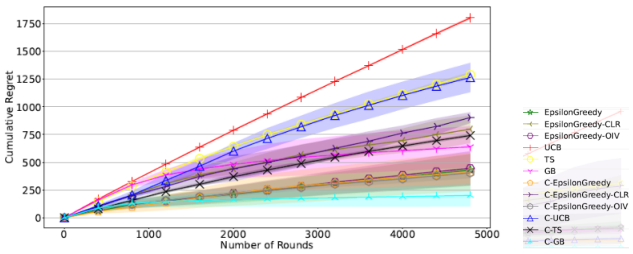

Fig. 3: Plot showing the evolution of regret with the number of pulls

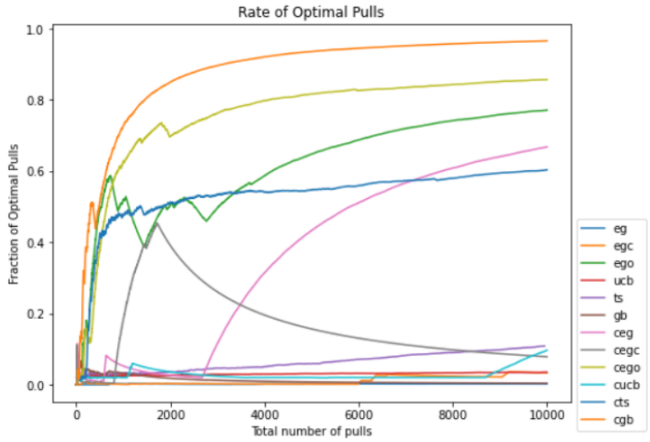

Fig. 4: The fraction of optimal pulls against the total number of arm pulls

Movie recommendation is a tougher task as compared to genre recommendation due to the larger number of available options (50). Thus, the results obtained on this differ slightly from that obtained on the genre recommendation, which had only 18 arms.
The observations from the plots are summarised as follows:

1) For all algorithms, the correlated versions out-perform the corresponding uncorrelated algorithms.
2) Once again, Gradient Bandit algorithms seem to give the best performance, both in terms of regret as well as number of optimal pulls.
3) Epsilon Greedy Algorithm with constant learning rate show areas where a very low number of pulls are attributed to the optimal arm.
4) Evidently, several of the algorithms are not able to determine the optimal arm, and thus have the fraction of optimal pulls tending to zero. Even these, however, do not have abnormally high regrets, suggesting that this settle for good arms, albeit sub-optimal. The algorithms unable to learn adequately include UCB, C-UCB, EGC, C-EGC, TS and C-TS.
5) Interestingly, C-EGC does pick up the right arm initially, but switches its selection mid-way. This may suggest discontinuity in dealing with empirical means, when a constant learning rate is used (as opposed to the natural definition of empirical means, which decrease as 1/n)

V. Variance and Accuracy

One important aspect of the algorithms analysed is the associated variance in them. For the probabilistic algorithms, the stream of pulls in each run differs, which can lead to slightly different regret trajectories. A higher variance of algorithms may also hamper their decisions, leading to sub-optimal outcomes in certain runs. For such algorithms, it is essential to take an ensemble to runs, and average the results out, to ensure reliability. The various degrees of variances can be seen int the regret plots in both : Genre and Movie Recommendation experiments.
Accuracy is another aspect which is closely related to variance, in that in a deterministic algorithm, running the instance once can give a fair idea of what the algorithm lack in, and ideas on how it can be improved. In probabilistic algorithms, on the other hand, an ensemble average may not be indicative of the strength of the algorithm, and different choices of hyper-parameters in these may lead to major differences in performance without any clear explanation.
This is the case with algorithms such as Epsilon-Greedy and Thompson Sampling. Further, the introduction of the correlated versions of the algorithms also strengthens the probabilistic nature of algorithms, leading to similar effects.

VI. Conclusions

In this project, we study the performance of six different algorithms, and their transformed versions as per [1], on a multi-armed bandit setting, where the arm reward distributions are corrrelated. We obtain experimental results on two real life datasets : Genre Recommendation and Movie Recommendation, and show how the proposed changes to algorithms enhance their performance in the correlated setting. Further, the performance of different algorithms are compared among themselves, using both regret-based analysis and using fraction of optimal pulls. The results and deductions are attached in the previous section.

References

[1] S. Gupta., S. Chaudhari, G. Joshi, O. Yagan : Multi-Armed Bandits with Correlated Arms
[2] Richard Sutton, Andrew Barto : Reinforcement Learning, An Introduction
[3] F. M. Harper and J. A. Konstan, “The movielens datasets: History and context,” ACM Transactions on Interactive Intelligent Systems (TiiS), vol. 5, 4, Article 19, 2015.
[4] https://towardsdatascience.com/thompson-sampling-fc28817eacb8

Appendix

Please find attached the number of pulls of each algorithm, based on a single run with a horizon of 5000 steps, on the genre recommendation dataset.
The actual means of each arm is also attached herewith. The highlighted row indicates the optimal arm.

Table 1: Sample Run on Genre Recommendation dataset

True Means	EG	EGC	EGO	GB	UCB	TS	CEG	CEGC	CEGO	CGB	CUCB	CTS
3.84	274	241	27	40	281	192	58	275	31	1	5	10
3.86	29	154	41	31	282	142	3	196	28	1	5	21
3.5	29	65	41	23	159	45	0	1	37	1	4	1
3.45	30	44	34	19	161	62	0	4	31	1	1	1
3.32	37	24	37	31	144	56	0	4	29	2	1	1
4.18	3973	1954	4441	4333	596	2378	4495	2948	4429	4819	4398	4327
3.86	23	305	33	38	328	127	2	358	33	1	4	31
3.81	23	144	43	45	290	109	0	198	42	5	3	1
3.68	22	71	40	37	216	83	0	97	40	2	3	8
3.75	33	59	26	38	242	112	8	36	37	4	4	8
3.75	36	112	32	39	249	166	0	5	39	5	4	11
3.65	31	46	27	22	193	100	0	11	24	1	1	1
3.69	33	41	31	35	237	103	1	67	31	1	1	2
3.59	30	48	36	37	218	87	1	46	35	1	1	1
3.99	300	330	26	54	397	324	191	303	43	55	278	278
4.03	32	514	20	84	476	679	221	291	33	94	278	275
3.68	33	71	32	36	253	69	0	7	29	1	1	1
3.82	32	77	33	58	278	166	2	109	29	5	8	13