PROJECT 4

Karishma Jain(kjain20)
Collaborator: Disha Jindal

Problem 1

The Multi-armed bandit problem is a problem where we are given k slot machines each having an arm and each arm has its own probability distribution of reward or it might not follow any distribution in case of non-stochastic setting. Pulling one of the arms gives us a reward of either 1(success) or 0(failure). Our objective is to maximize the total reward in the long run. In the dataset given to us, there are 50 different category of ads which is equivalent to k in the multi-armed bandit problem. A user is shown one of those category in each round (there are total of 32657 rounds) and depending upon whether the user clicks on the category of ad chosen (equivalent to pulling a arm in the multi-armed bandit problem), we get a reward. If the user clicks on the ad shown to him, then we get a reward of 1(loss is 0) and if he does not click, then we get a reward of 0(loss is 1).

We analyzed a number of algorithms such as UCB, Thompson Sampling and EXP3 to solve this problem. Following are the observations after during initial analysis:

1. In Thompson Sampling as well as UCB we assume that the rewards are coming from a probability distribution. But we do not have any such assumption in EXP3. Since, we think that it is not a right assumption to make in this scenario, we think EXP3 is a better fit in an adversarial setting like this.
2. Another observation is that UCB is deterministic in nature but Thompson Sampling and EXP3 are not. There is no randomness in the decision making in case of UCB and it can be exploited by the adversary as the next data point based on the user might not like the same ad the next time which had got the reward earlier.

Based on these observations, EXP3 looks like a good fit for this setting. But since, Thompson sampling is also stochastic in deciding the next arm and we are not sure whether the rewards follow any distribution or not. We will explore and implement Thompson Sampling and EXP3 in partial and full feedback settings.

Metrics:

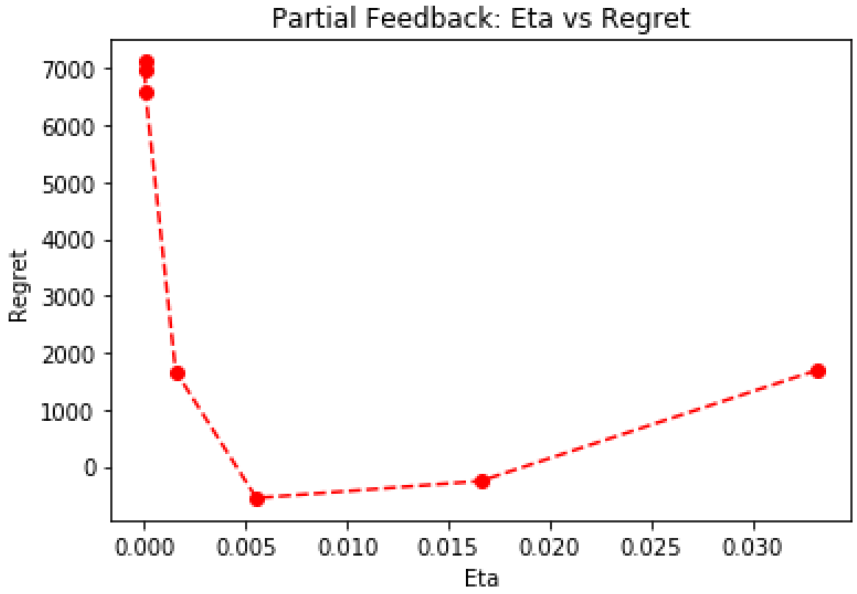
In EXP3, we are comparing the partial loss of our algorithm with that of the optimal row wise loss. Although for analysis purpose, we have plotted column wise loss at each point.
For Thompson Sampling, we are comparing the partial loss of our algorithm with that of the optimal row wise(best arm till that round) loss.

Problem 2 :

EXP3: Implemented EXP3 to solve the given multi arm bandit problem with partial feedback.

Regret Analysis: We ran the algorithm for different values of eta to analyze the behavior of eta vs regret. Following graph shows this relationship:

Eta	Regret
6/sqrt(t)	1690
3/sqrt(t)	-259
1/sqrt(t)	-558
sqrt(ln k / tk)	1671
2/(t + 1)	6573
1/(t + 1)	6980
1/2(t + 1)	7119



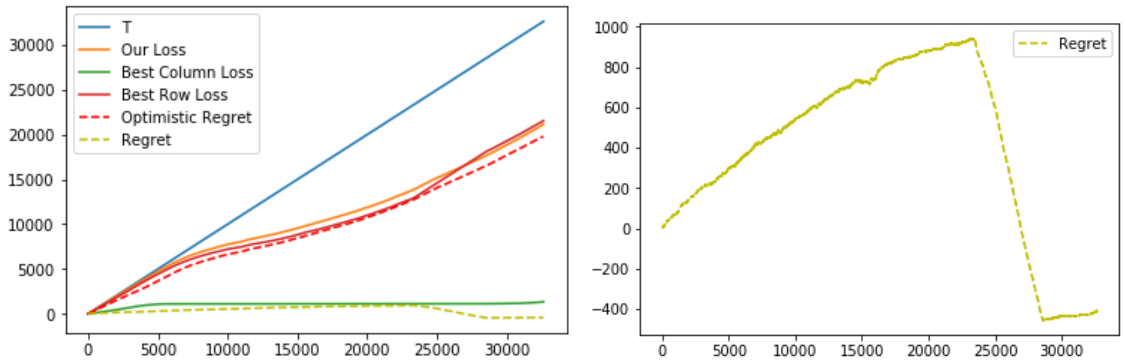
From the table we can see that we got the best regret of -558 at eta = 1/sqrt(t). As we can observe from the above plot, as we increase eta, the regret decreases upto an optimal eta and then the regret starts increasing again. During the initial phase when the eta is small the probability update at each step is less and thus, there is more of exploration and less of exploitation. In the second phase of the graph, the regret starts increasing after reaching an optimal regret, since the eta is high so it is changing the probability distribution by a larger extent and hence, it is getting skewed towards the arms which were giving rewards till now. It in turn will make the probability of choosing such arms a lot higher leading to very less exploration and thus, the regret starts increasing again as we are exploited too much at sometime and now the user might not be clicking on the same ads leading to increase in regret.

Individual Plots for different values of eta

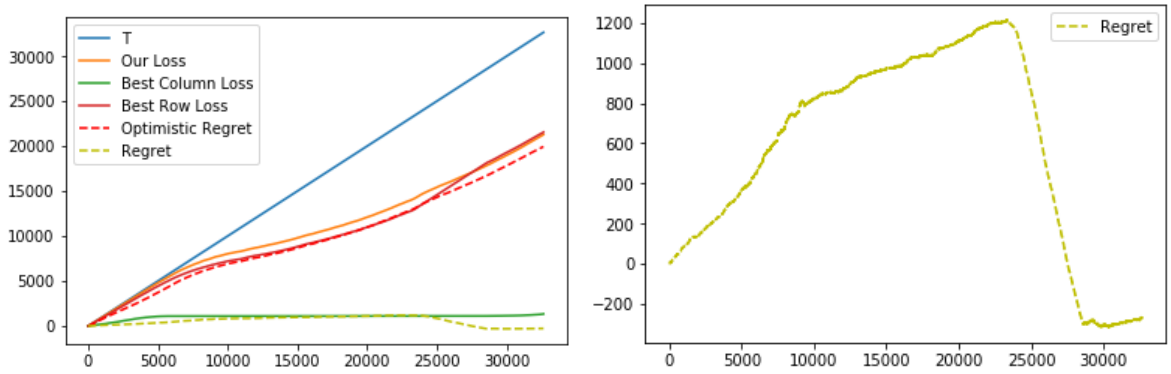
Showing detailed visualization of regret, best row-wise loss, best column-wise loss

In the below plots, we have plotted T just to see whether our regret plot is sublinear. We have plotted our loss which is comparable to the best row loss. We have plotted the best column loss just for our reference for getting the idea of the optimal regret. On the right side, we have plotted the magnified version of our regret. The below are the plots where x-axis is T(number of rounds) and y-axis is regret.

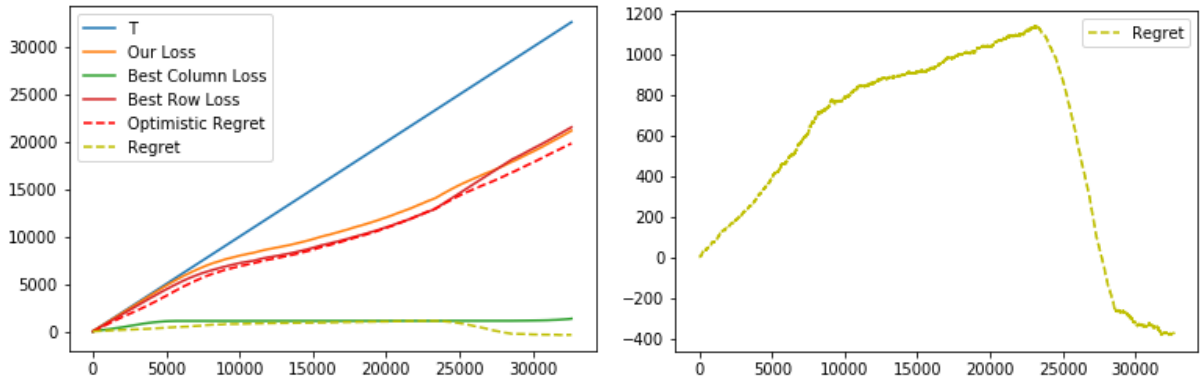
6/sqrt(t) :



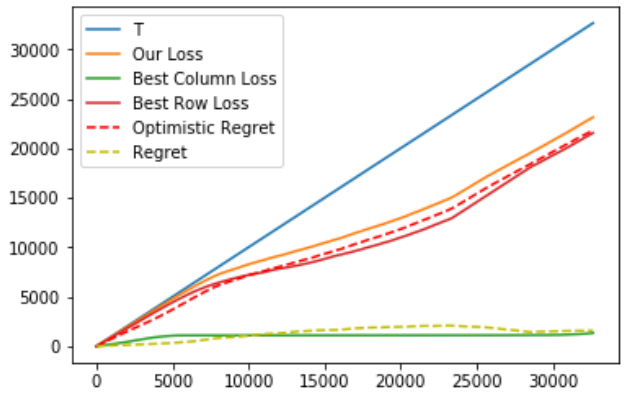
3/sqrt(t) :

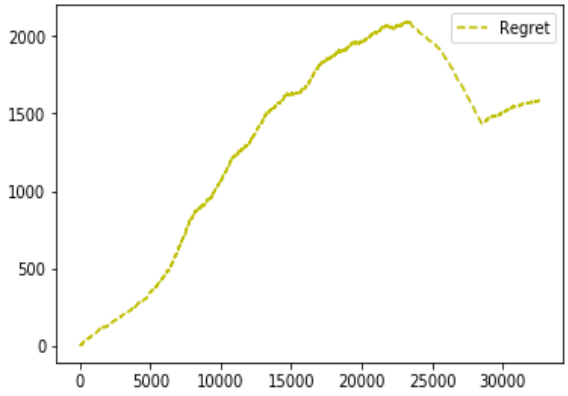


1/sqrt(t) :

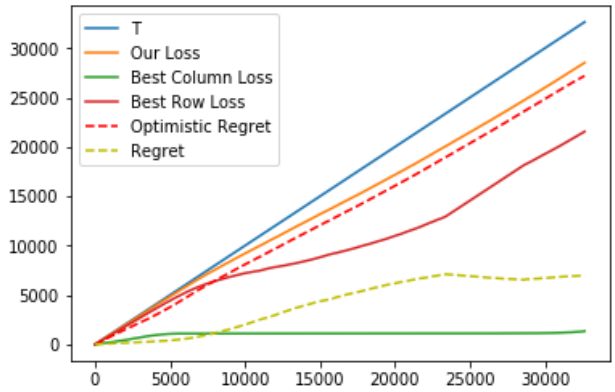


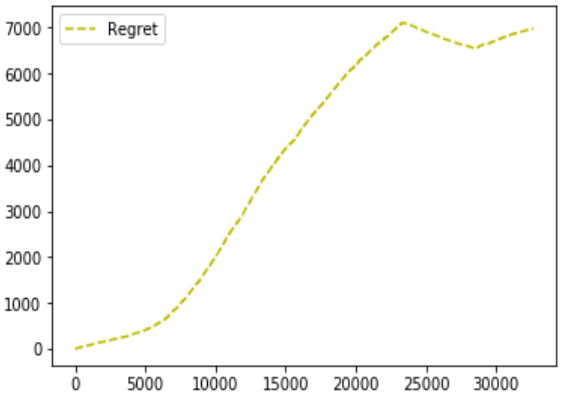
sqrt(ln k / tk) :



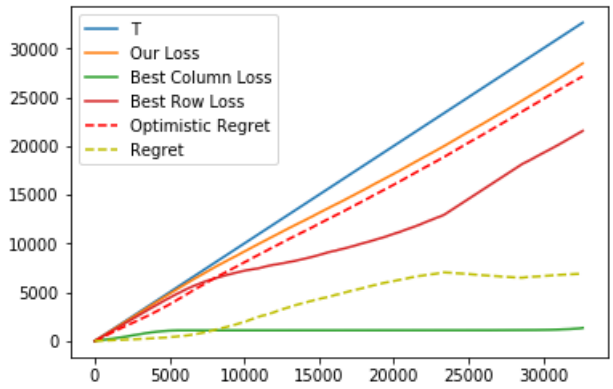


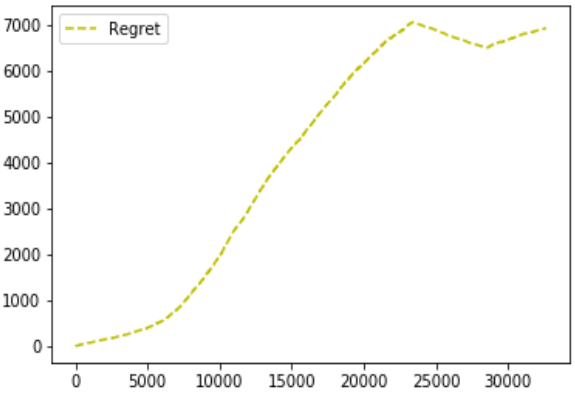
2/(t + 1) :



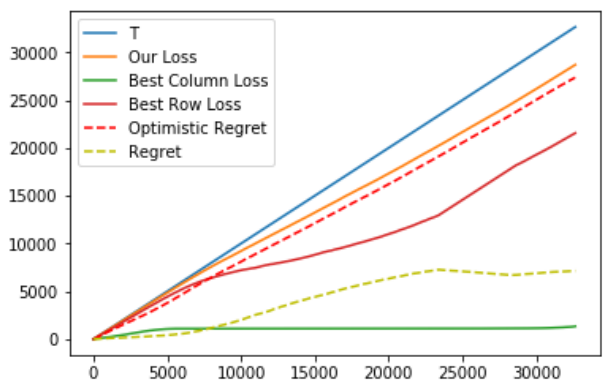


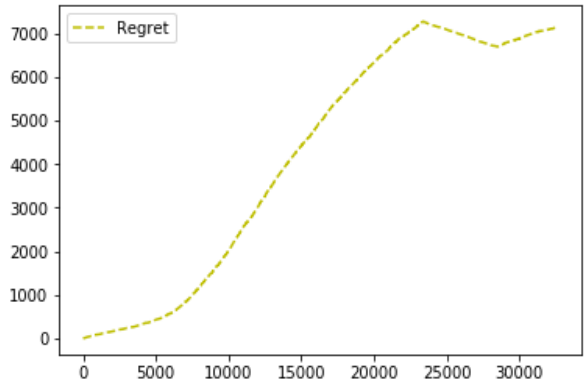
1/(t + 1) :





1/2(t + 1) :





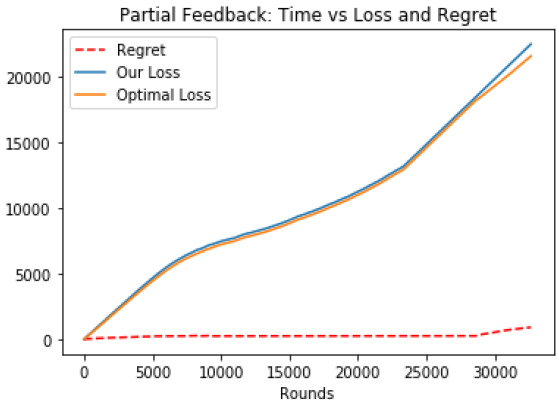
Analysis for the individual plots:

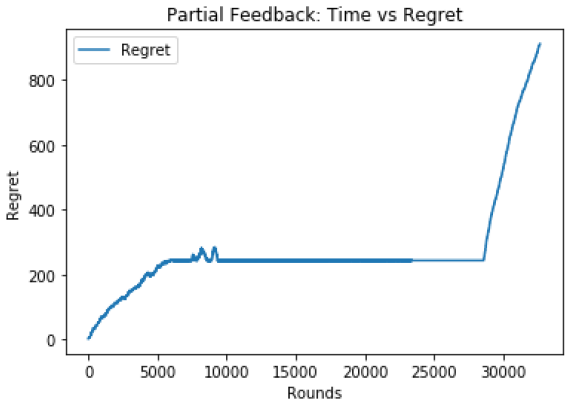
As we can observe from the regret plot, there comes a point when there is huge dip in the value of the regret, that's when the algorithm is exploiting more but we need to explore as well sometimes because the user might click on the ad which he did not click on when he was shown previously. Such an ad should be assigned less probability but should not be made 0 because we don't want to show it often but in the future we do want to explore by showing that ad again and check whether the user's interest might have changed. When eta is high, we are giving too low scope for exploration in the future which is also bad as we can observe from the above plot that the regret starts increasing but at a low rate. There is always a trade-off between exploration and exploitation in the sequential decision problem. As we can observe from the below plots of regret for different values of eta, as eta decreases the dip also decreases and the graphs become smoother but for too small value of eta, the regret is high because there is lot more exploration then it should have been. So somewhere we want to select eta which is not too high neither too low.

Thompson Sampling:

The Thompson Sampling algorithm initially assumes arm i to have prior Beta(1, 1) on mean_i , which is the uniform distribution on (0, 1). At time t, having observed Si(t) successes (reward = 1) and Fi(t) failures (reward = 0) in Si(t) + Fi(t) plays of arm i, the algorithm updates the distribution on mean_i as Beta(Si(t) + 1, Fi(t) + 1). The algorithm then samples from these posterior distributions of the μi's, and plays an arm according to the probability of its mean being the largest.

Below, we are plotting the graph for our loss, optimal loss and regret. On the right side, we have plotted the magnified version of our regret.





Regret: 911
Mean Regret: 0.0279

Analysis:
In Thompson's sampling, the beta distribution is updated only for the arm that is picked and if there is a reward of 0, the beta distribution for that arm gets a bit of left skewed and if there is a reward of 1, the beta distribution for that arm gets a bit of right skewed. But the beta distribution for the arm that is not picked remains unchanged and hence initially the algorithm explores a lot. As we can observe from the below graph of regret, the regret remains stable for quite long. This is because a good trade-off is achieved between exploration and exploitation since the distribution gets skewed slowly. But after sufficient number of rounds, the beta distribution of the ads that performed well (user clicked the ad) is quite right skewed and the once that didn't perform well is quite left skewed so there will be much more exploitation than should have been as sample from these distribution will be higher most of the times so the same kind of ads will be showed but we don't want to do that. We also want to do less but sufficient exploration because we also want to check if the user's interest might have changed then we do want to increase the probability of showing that add which the user did not click before but can click now depending on his changed interest. That is why the regret starts increasing after remaining stable for a good amount of time as the shift in the beta distribution of each arm is slow.

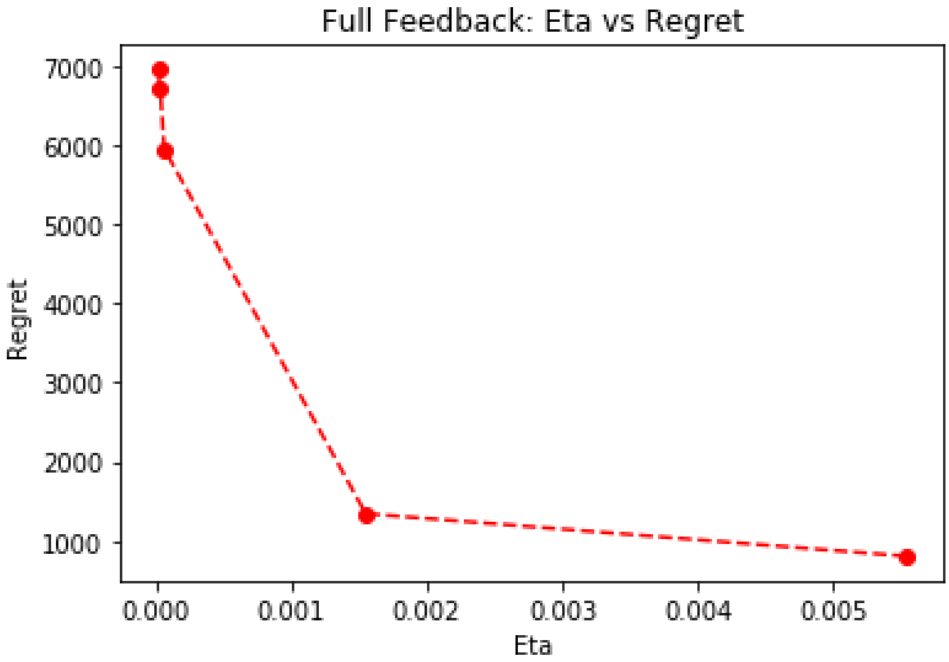
As we can see, from the numbers EXP3 performed much better than Thompson Sampling. Although, we observed some issues with EXP3 which are discussed in more detail in Problem 4 where we have provided a resolution for the mentioned problems and improved the regret further.

Problem 3: Implemented EXP3 to solve the given multi arm bandit problem with full feedback.

Regret Analysis: We ran the algorithm for different values of eta to analyze the behavior of eta vs regret. Following table and graph shows the results and the relationship:

Eta	Regret
1/sqrt(t+2)	808

sqrt(ln k / tk)	1344
2/(t + 2)	5949
1/(t + 2)	6719
1/2(t + 2)	6956

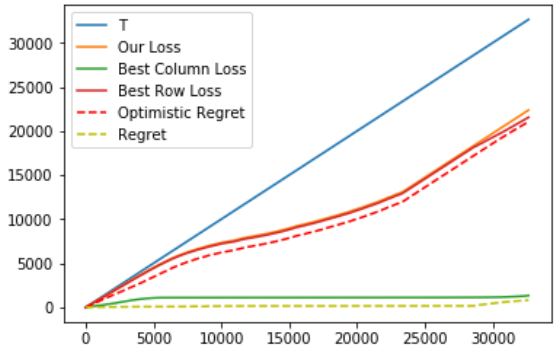


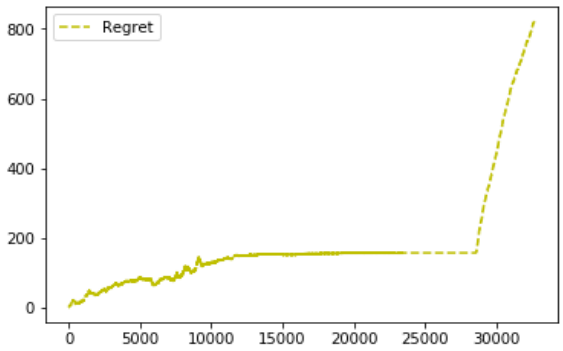
The overall trend of full feedback is similar as that of partial feedback where the regret decreases initially and once it reaches the optimal point, it starts increases with the increase in eta. The plots are even steeper in case of full feedback because in case of partial feedback, we used to get feedback from only one arm that is picked (one ad that is shown) whereas in this case, since we get feedback for all the arms, the probability distribution gets skewed even faster. For the same values of eta, the update in case of full feedback is way more as compared to partial. Because of this over exploitation the regret shoots up a lot for higher values of eta. So, from this and more clearly from the table we can see that we got the best regret of 808 at eta = 1/sqrt(t).

Individual Plots for different values of eta

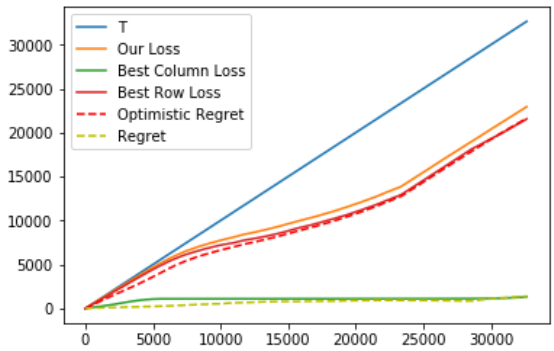
Showing detailed visualization of regret, best row-wise loss, best column-wise loss

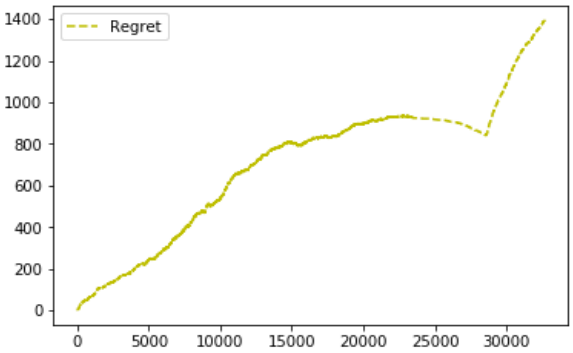
1/sqrt(t) :



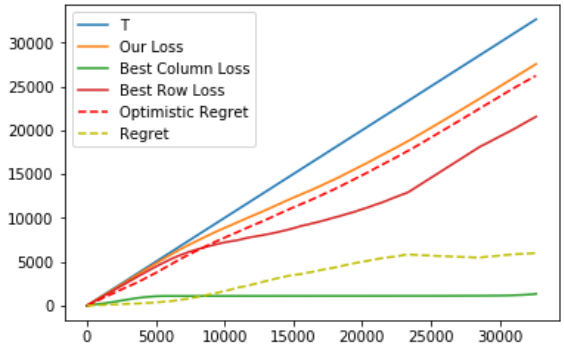


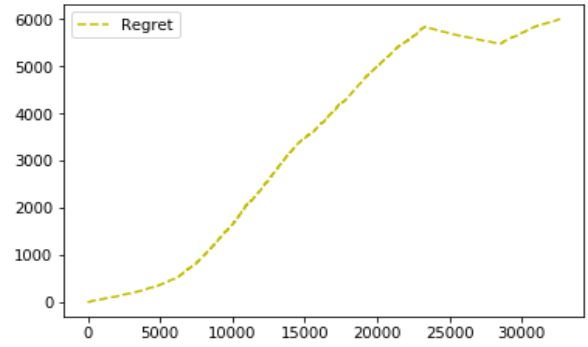
sqrt(ln k / tk) :



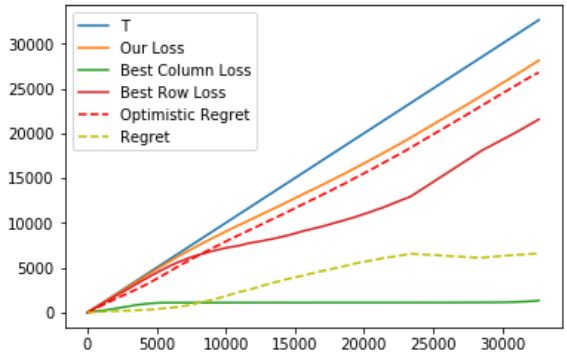


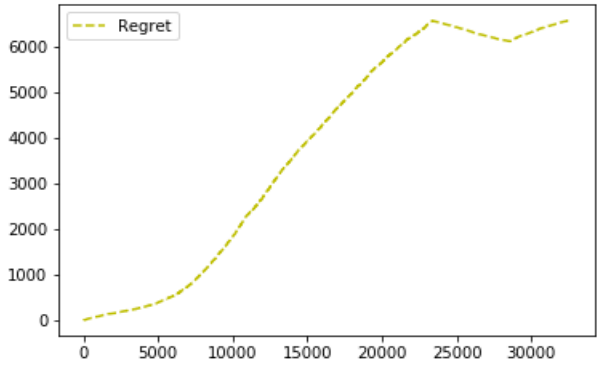
2/(t + 1) :



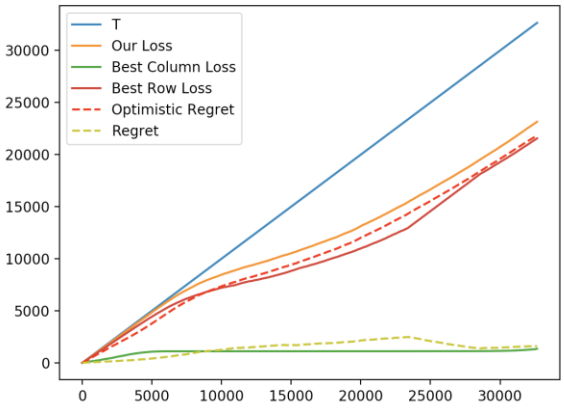


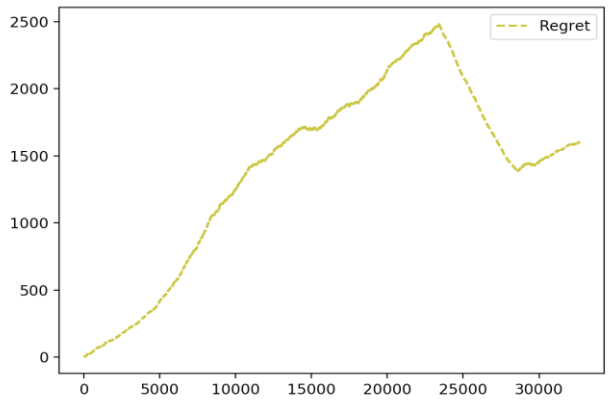
1/(t + 1) :





1/2(t + 1):

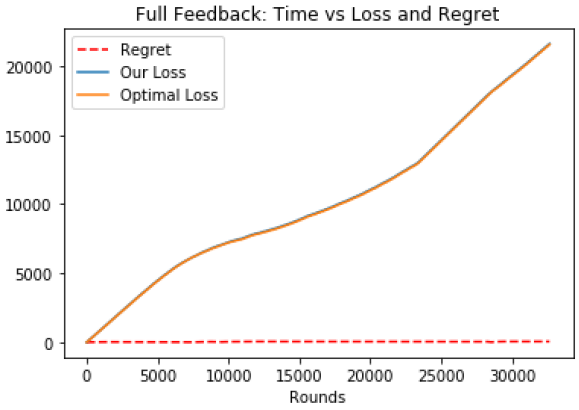




Analysis of the individual plots of the regret :

As we can observe from the above plots, when the value of eta is high, there is a gradual increase in the regret when there is a good trade-off maintained between exploration and exploitation but since high eta, penalizes the probability of ads that the user did not click more, probabilities of many of the arms will become small and the ads that the user clicked on will have very high probabilities. So in case of high eta, there is strong exploitation after a while but we do want to explore a little in between, hence there is a steep increase in the regret. But as eta decreases, the increase in regret is much more smoother. But in case of very low eta, the probabilities of bad ads(the one that user didn't click) won't become very low because of which there is no steep increase in regret but because of very less exploitation, the regret is higher. Hence, we chose eta which are somewhere in between these extremes.

Thompson Sampling:



Regret: 58
Mean Regret: 0.001776

Initially, there will more of exploration, so the regret increases as can be seen from the above graph. In case of full feedback, the beta distribution of all the arms will be updated in each round. In the phase where the regret becomes stable, the algorithm has achieved a good trade-off between exploration and exploitation. After good enough number of rounds, the beta distribution for every arm will either be right skewed (an ad that the user clicked on) or left skewed (an ad that the user did not click on), and there will be more of exploitation and less of exploration. But as we can observe there is a dip in the graph. In case of exploitation, same kind of ads are being showed most of the times. It could be so that the user still is interested and clicks on the ads shown, so the regret becomes less. But showing the same kind of ads continuously for a long period wouldn't work as the user's interest might change so we do want to keep on exploring as well to get an update on that. That is why the regret starts increases.

Problem 4

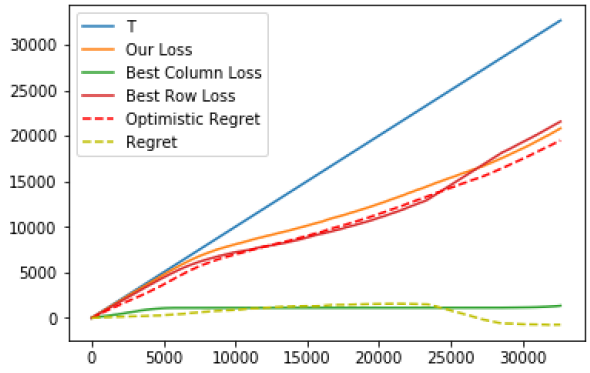
In problem 2, we discussed about Thompson Sampling and EXP3 in detail for partial feedback. We observed that EXP3 performed better with giving the best regret of -558 at eta = 1/sqrt(t). Although Exp3 performed well and gave a very small regret, there are some issues with EXP3 which can improved by tweaking few things.

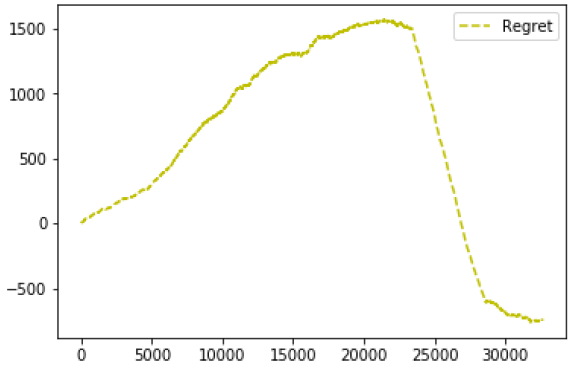
EXP3 is designed in such a way that it chooses the best arm at each round by taking care of both exploration and exploitation. It does so by maintaining a probability distribution starting with a uniform distribution, using it to decide randomly which action to take next, and increasing (decreasing) the relevant weights when a payoff is good (bad). But, sometimes it does not explore enough as it concentrates the probabilities on the wrong ad or to say the the ad which is no longer liked by users, for too long and then incurs a large regret.

To solve this problem, we have added a constant factor to the probability distribution so as to ensure a lower bound on the probability of exploration. We have introduced a parameter gamma which belongs to [0,1); which tunes the desire to pick an action uniformly at random. That is, if gamma = 1, the weights have no effect on the choices at any step. And if the gamma = 0, it is same as the existing EXP3 algorithm. So, the value of gamma should lie in between these extremes.

P_EXP3 = Probability distribution as per EXP3
P = (1- gamma)*P_EXP3 + gamma/k

EXP3 gave us the best results for eta = 1/sqrt(t), where t is the number of iterations. Since, eta is between 0 to 1. We chose the value of gamma to be equal to eta. After implementing this change, we analyzed the performance of EXP3 with and without gamma.





Eta	Regret without gamma	Regret with gamma
1/sqrt(t)	-558	-975

Conclusion:
For Partial Feedback: Exp3 + Uniform distribution for exploration
Regret: -975
Mean Regret: - 0.029764

For Full Feedback: Thompson Sampling
Regret: 58
Mean Regret: 0.001776

For full feedback, Thompson Sampling gave the better results as compared to EXP3. In case of full feedback, we update the beta distribution of all arms. It leads to more update in one step and so, the rewards (expected mu values) will tend to converge to their actual mu value(assuming that the rewards do come from a probability distribution) faster in case of

Thompson sampling as compared to the partial feedback. So, if the data satisfies the probability assumptions, the regret bounds of Thompson sampling is O(log T) which is better than EXP3 which is O(sqrt(T)). We observed such a behavior in the full feedback case and got better convergence for Thompson sampling.

We observed that, EXP3 + uniform distribution worked better for partial feedback as compared to EXP3 and Thompson Sampling. In case of EXP3, we choose the arm based on a common probability distribution and updated the same distribution in each step. Whereas, in case of Thompson Sampling we maintain k beta distributions and update one of them in each step of partial feedback loop. Since, we only change one distribution in case of Thompson, the updates are not influencing the other arms and hence, makes the overall progress very slow. Since, we update the common distribution in EXP3, even though we get the feedback for one arm, we change the probability(since, the denominator changes) of all the arms at each step. Because of this reason, we start exploiting relatively faster in case of EXP3 and after adding the constant distribution for exploration, our performance has increased even more because we have fixed the no exploration problem that EXP3 faces at times.