Leveraging Preference Data to Improve Online Learning

EE641 Project Report

Kranti Adsul
kadsul@usc.edu

Akhil Agnihotri
akhil.agnihotri@usc.edu

Nikita Dalvi
dalvin@usc.edu

Abstract

In this project, we wish to investigate the extent to which offline demonstration data can improve online learning. It is natural to expect some improvement, but the question is how, and by how much? We plan to characterize the quality of demonstration data to generate portable insights as we focus on Thompson sampling (TS) applied to a multi-armed bandit as a prototypical online learning algorithm and model. The demonstration data shall be generated by a (expert) rater with a given 'competence' level. Based on this theoretical algorithm, we plan to conduct experiments, if possible, or then develop a practical TS algorithm that utilizes the demonstration data in a coherent. This shall offer insight into how pretraining can greatly improve online performance as is seen currently in the Reinforcement Learning with Human Feedback (RLHF) paradigm, or in the recommender system space.

1 Introduction

A modern paradigm for developing intelligent agents involves pretraining on large quantities of existing data followed by learning from real-time interactions. For instance, to produce a chatbot, one can pretrain a large language model on text gathered from the internet and subsequently improve behavior through learning from interactions with humans [Ziegler et al., 2019, Ouyang et al., 2022]. With such an approach, the pre-existing text is treated as offline demonstration data that conditions a reinforcement learning agent before it engages in online learning. It is natural to expect the offline demonstration data to improve performance of the online learning agent in unknown environments. However, the degree of improvement must depend on the quality of the demonstration data. If the data is produced by a *competent* expert, it ought to improve the agent's performance more so than if not.

As a prototypical model for online learning, we consider multi-armed bandits that offer a simple context for understanding the role of offline data, in addition to a simple model for data generation. We focus on Thompson sampling (TS) [Thompson, 1933] which is a popular online learning algorithm, owing to its effectiveness across a wide range of environments. Hence, we develop our warm Thompson Sampling (wTS) algorithm that incorporates the given offline dataset in Thompson sampling.

This problem is relevant to the context of Reinforcement Learning with Human Feedback (RLHF) or recommendation systems. Firstly, it enables the agent to efficiently explore a vast action space and learn from historical user interactions, enhancing recommendation accuracy. Secondly, it allows for the incorporation of diverse sources of feedback, such as user preferences and expert demonstrations, leading to more robust and adaptive recommendation systems.

Related Work. There is a rich body of literature on learning algorithms for bandits (see Russo et al. [2018], Lattimore and Szepesvári [2020] for a detailed review). Almost all of this literature assumes that the learning agent starts from scratch but this may lead to a long initial learning stage. In fact, offline data is available for many applications, such as training a large language model Ouyang et al. [2022].

In the context of reinforcement learning (RL), there are several recent works [Rashidinejad et al., 2021, Xie et al., 2021, Wagenmaker and Pacchiano, 2022] that bridge offline and online RL. However, all of them focus on policy optimization rather than regret minimization. And they require different versions of concentrability coefficient conditions that are hard to be satisfied in practice.

2 Preliminaries

As stated before, we plan to work within the K-armed bandit setting, so consider a stochastic K-armed *linear* bandit problem.

Linear bandits are a class of multi-armed bandit problems where arms represent choices or actions, and each arm's reward is modeled as a linear function of its features. Let $A_t \in \mathbb{R}^d$ denote the feature vector of the chosen arm at time t, and R_t represent the observed reward. The objective is to maximize cumulative rewards over a finite time horizon T.

In this setup, we have a set of K actions, $\mathcal{A} = \{a_1, \ldots, a_K\} \subseteq \mathbb{R}^d$. The environment is characterized by a random vector $\theta \in \mathbb{R}^d$, with an unknown prior distribution ν_0. At each time step t, the agent chooses an action $A_t \in \mathcal{A}$ and receives a reward R_t:

$$R_t = \langle A_t, \theta \rangle + \eta_t$$

where $\eta_t \sim N\left(0, \sigma^2\right)$. In addition, we also have an initial dataset $\mathcal{D}_0$, which is generated by a human rater. The offline dataset is of the form:

$$\mathcal{D}_0 = \left\{ \left(\overline{A}_n^{(0)}, \overline{A}_n^{(1)}, Y_n \right) \right\}_{n=1}^{N},$$

where $\overline{A}_n^{(0)}, \overline{A}_n^{(1)} \in \mathcal{A}$ are two actions, and $Y_n \in \{0, 1\}$ indicates the rater's preference. In particular, $Y_n = 0$ indicates that the rater prefers action $\overline{A}_n^{(0)}$ to $\overline{A}_n^{(1)}$, and $Y_n = 1$ indicates vice versa. We assume that the offline dataset is characterized by the following parameters:

- the dataset size N.
- the action sampling distribution μ, where $\overline{A}_n^{(0)}$ and $\overline{A}_n^{(1)}$ are i.i.d. sampled from μ (for now we consider i.i.d. as part of the minimum viable solution).
- the rater's competence, in particular its knowledgeability λ and deliberateness β. We assume that the rater's knowledge takes the form of a vector ϑ, which is distributed as $\vartheta \sim N\left(\theta, \mathbf{I}/\lambda^2\right)$. Given two actions $\overline{A}^{(0)}$ and $\overline{A}^{(1)}$, the rater chooses action $\overline{A}^{(0)}$ with probability

$$P\left(Y = 0 \mid \overline{A}^{(0)}, \overline{A}^{(1)} \; ; \; \theta\right) = \frac{\exp\left(\beta \left\langle \overline{A}^{(0)}, \vartheta \right\rangle\right)}{\exp\left(\beta \left\langle \overline{A}^{(0)}, \vartheta \right\rangle\right) + \exp\left(\beta \left\langle \overline{A}^{(1)}, \vartheta \right\rangle\right)} \tag{2.1}$$

Then, if the initial offline dataset is $\mathcal{D}_0$ and the online collected dataset collected upto and including time t as $\mathcal{H}_t = \{(A_t, R_t)\}_{s=1}^{t}$, then at any time step t, the dataset available is $D_t = \mathcal{D}_0 \cup \mathcal{H}_{t-1}$.

Problem statement. Given an offline preference dataset $\mathcal{D}_0$ and an environment available for the online phase for T rounds, minimize the Bayesian Regret as given by:

$$\mathrm{BR}_T^{\pi} := \sum_{t=1}^{T} \mathbb{E}_{\pi} \left[\langle A^\star, \theta \rangle - R_t \right] \tag{2.2}$$

, where $A^\star = \operatorname{argmax}_{a \in \mathcal{A}} \langle a, \theta \rangle$, and π is a policy, which is a distribution over the action space simplex.

3 warm Thompson Sampling (wTS)

We plan to use the above setting to construct the warm Thompson Sampling algorithm. First some assumptions:

Assumption 1. *The environment vector θ has an initial Gaussian distribution i.e. $\nu_0(\theta) \sim \mathcal{N}(\mu_0, \Sigma_0)$.*

Assumption 2. *The action sampling distribution for constructing the offline dataset $\mathcal{D}_0$ is denoted by μ, where $\overline{A}_n^{(0)}$ and $\overline{A}_n^{(1)}$ are i.i.d. sampled from μ.*

The algorithm will be as follows:

1. Using the offline dataset $\mathcal{D}_0$, construct an informed prior $P(\theta \mid \mathcal{D}_0)$ as follows:
$$\nu_{\text{off}}(\theta) = P(\theta \mid \mathcal{D}_0) \propto P(\mathcal{D}_0 \mid \theta) \cdot \nu_0(\theta)$$
2. At time t, the environment parameter θ has prior distribution $\nu_{t-1}(\theta)$, and the collected online data is $\mathcal{H}_{t-1}$. Using $\mathcal{D}_t = \mathcal{D}_0 \cup \mathcal{H}_{t-1}$ with $\nu_1(\theta) = \nu_{\text{off}}(\theta)$, we update our posterior for $t \geq 2$ to get a better estimate of the environment using $\nu_t(\theta)$.
3. At time t, observe the reward R_t after selecting action A_t as $A_t = \text{argmax}_{i \in A} \langle A_i, \theta_t \rangle$, where $\theta_t \sim \nu_t(\theta)$.

3.1 Dataset

Given the distributions of the offline data in Section 2, data generation follows easily and is not compute intensive. Unfortunately, we don't have any benchmark datasets available for this problem setting as it has not been studied in the past before.

4 Expected Results

By the end of this project, we aim to have successfully studied how the quality of the offline data impacts online learning (for example, by varing the knowledgeability λ and deliberateness β), which is a current unsolved problem in the context of RLHF and recommendation systems. We expect to see improvement in the online learning phase as we increase λ.

The challenges we hope to face are in the updating of the posterior in the wTS algorithm, due to loss of conjugacy. In that case, we aim to come up with a surrogate Maximum A Posteriori (MAP) estimate of the posterior.

Questions.

1. Why can't we use an informed Bernoulli / Binomial distribution instead of the softmax for modeling the rater's preference? Thompson sampling originally uses the Beta distribution which is always within [0,1] and has nice conjugacy with Binomial distribution (if Binomial is used above instead of Bernoulli). Why use Normal here?

   $\hookrightarrow$ Can do but using Bernoulli for arms leads to increase in parameters $(= 2K)$, but with linear bandits we have fixed size since the environment parameter is independent of number of arms. Also, Beta distribution makes sense for discrete action space, here $\mathcal{A} \subset \mathbb{R}^d$, which is continuous $\Rightarrow$ infinite parameters for infinite actions.
2. Why not use Gibbs sampling?

   $\hookrightarrow$ Bayesian posterior sampling captures the maximum information given a dataset, and hence can be treated as a sufficient statistic of the offline dataset.

5 Theoretical Thompson Sampling

This algorithm uses the offline preference data as described below. First some assumptions.

Assumption 3. *The environment vector θ has an initial Gaussian distribution, i.e., $\nu_0(\theta) \sim \mathcal{N}(\mu_0, \Sigma_0)$.*

Assumption 4. *The action sampling distribution for constructing the offline dataset $\mathcal{D}_0$ is denoted by μ, where $\overline{A}_n^{(0)}$ and $\overline{A}_n^{(1)}$ are i.i.d. sampled from μ.*

5.1 Warm-starting the online phase

Using the offline dataset $\mathcal{D}_0$, construct an informed prior $P(\theta \,|\, \mathcal{D}_0)$ as follows:

$$\begin{aligned}
\nu_{\text{off}}(\theta) = P(\theta \,|\, \mathcal{D}_0) &\propto P(\mathcal{D}_0 \,|\, \theta) \cdot \nu_0(\theta) \\
&\propto \left[\prod_{n=1}^{N} P(Y_n \,|\, \overline{A}_n^{(0)}, \overline{A}_n^{(1)}, \theta) \cdot P(\overline{A}_n^{(0)} \,|\, \theta) \cdot P(\overline{A}_n^{(1)} \,|\, \theta) \right] \cdot \nu_0(\theta) \\
&\propto \left[\prod_{n=1}^{N} \frac{\exp\left(\beta \langle \overline{A}^{(Y_n)}, \vartheta \rangle\right)}{\exp\left(\beta \left\langle \overline{A}_n^{(0)}, \vartheta \right\rangle\right) + \exp\left(\beta \left\langle \overline{A}_n^{(1)}, \vartheta \right\rangle\right)} \right] \cdot \nu_0(\theta) \\
&\propto \frac{\exp\left(\beta \left\langle \sum_{n=1}^{N} \overline{A}^{(Y_n)}, \vartheta \right\rangle\right)}{\prod_{n=1}^{N} \left[\exp\left(\beta \left\langle \overline{A}_n^{(0)}, \vartheta \right\rangle\right) + \exp\left(\beta \left\langle \overline{A}_n^{(1)}, \vartheta \right\rangle\right) \right]} \cdot \nu_0(\theta)
\end{aligned} \tag{5.1}$$

where, the second step follows from Equation (2.1) and the i.i.d. assumption of $\overline{A}_n^{(0)}$ and $\overline{A}_n^{(1)}$, and the third step from the absorption of the denominator of Equation (2.1) and the $1/K^2$ factor by the proportional sign, where K is the number of arms. $\overline{A}_n^{(0)}$ and $\overline{A}_n^{(1)}$ only depend on the (uniform) action sampling distribution μ and are independent of the environment parameter θ.

[Akhil: Sequence: Construct new estimate of $\theta \rightarrow$ take action]

5.2 Updating knowledge of the environment

At time t, the environment parameter θ has distribution $\nu_t(\theta)$, and the collected online data is $\mathcal{H}_{t-1}$. Using $\mathcal{D}_t = \mathcal{D}_0 \cup \mathcal{H}_{t-1}$ with $\nu_1(\theta) = \nu_{\text{off}}(\theta)$ and $\mathcal{H}_0 = \{\}$, we update our posterior for $t \geq 2$ as,

$$\begin{aligned}
\nu_t(\theta \,|\, \mathcal{D}_t) &\propto P(\mathcal{D}_t \,|\, \theta) \cdot \nu_0(\theta) \equiv P(\{(A_t, R_t)\} \,|\, \mathcal{D}_{t-1}, \theta) \cdot \nu_{t-1}(\theta \,|\, \mathcal{D}_{t-1}) \\
&\propto P(R_t \,|\, A_t, \theta) \cdot P(A_t \,|\, \mathcal{D}_{t-1}) \cdot \nu_{t-1}(\theta \,|\, \mathcal{D}_{t-1}) \\
&- - - - - - - - - - \\
&\propto \left[\prod_{s=1}^{t-1} P(R_s \,|\, A_s, \mathcal{D}_{s-1}, \theta) \cdot P(A_s \,|\, \mathcal{D}_{s-1}, \theta) \right] \cdot \nu_1(\theta \,|\, \mathcal{D}_0) \\
&\propto \left[\prod_{s=1}^{t-1} P(R_s \,|\, A_s, \theta) \cdot P(A_s \,|\, \mathcal{H}_{s-1}) \right] \cdot \nu_1(\theta \,|\, \mathcal{D}_0)
\end{aligned} \tag{5.2}$$

In addition, the posterior of ϑ also changes, and hence,

$$\vartheta_t \sim \mathcal{N}(\theta_t, \mathbf{I}/\lambda^2), \quad \text{where} \;\; \theta_t \sim \nu_t(\theta). \tag{5.3}$$

Note. It is worthwhile to mention that ϑ is only sampled initially while creating the offline dataset $\mathcal{D}_0$. After $\mathcal{D}_0$ is generated, rater's knowledge ϑ has no effect on the online phase of learning. However, as we will see, it is easier to calculate the joint posterior (θ_t, ϑ_t) than just θ_t.

5.3 Online decision making

At each time step t, observe the reward R_t after selecting action A_t as $A_t = \operatorname{argmax}_{a \in A} \langle a, \theta_t \rangle$ and update the dataset as $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{A_t, R_t\}$.

6 Practical Thompson Sampling

Since the posterior is intractable and loss of conjugacy occurs, we develop a ‘dual’ optimization problem, whose solutions $(\theta^\star, \vartheta^\star)$ are the point MAP estimates.

6.1 Surrogate Loss Function

$$\begin{aligned}
\underset{\theta,\vartheta}{\operatorname{argmax}}\ P(\theta,\vartheta \,|\, \mathcal{D}_t) &= \underset{\theta,\vartheta}{\operatorname{argmax}}\ P(\mathcal{D}_t \,|\, \theta,\vartheta)\cdot P(\theta,\vartheta) \\
&= \underset{\theta,\vartheta}{\operatorname{argmax}}\ \ln P(\mathcal{D}_t \,|\, \theta,\vartheta) + \ln P(\theta,\vartheta) \\
&= \underset{\theta,\vartheta}{\operatorname{argmax}} \underbrace{\ln P(\mathcal{H}_{t-1} \,|\, \mathcal{D}_0,\theta,\vartheta)}_{\mathcal{L}_1} + \underbrace{\ln P(\mathcal{D}_0 \,|\, \theta,\vartheta)}_{\mathcal{L}_2} + \underbrace{\ln P(\theta,\vartheta)}_{\mathcal{L}_3}
\end{aligned} \tag{6.1}$$

Then,

$$\begin{aligned}
\mathcal{L}_1 &= \sum_{s=1}^{t-1} \underbrace{\ln P(A_s \,|\, \mathcal{D}_{s-1},\theta,\vartheta)}_{\text{indep. of } \theta,\vartheta \ \Rightarrow\ \text{const}} + \ln P(R_s \,|\, A_s,\theta,\vartheta) \\
&= \text{const} \ - \ \frac{t-1}{2}\ln\left(\frac{2\pi}{\sigma^2}\right) - \frac{1}{2}\sum_{s=1}^{t-1}\big(R_s - \langle A_s,\theta\rangle\big)^2. \\
\mathcal{L}_2 &= \sum_{n=1}^{N} \ln\left(\left(\overline{A}_n^{(0)}, \overline{A}_n^{(1)}, Y_n\right) \,\Big|\, \theta,\vartheta\right) \\
&= \sum_{n=1}^{N} \ln\left(Y_n \,\Big|\, \overline{A}_n^{(0)}, \overline{A}_n^{(1)}, \theta,\vartheta\right) + \quad \underbrace{\ln P\left(\overline{A}_n^{(0)}, \overline{A}_n^{(1)} \,\Big|\, \theta,\vartheta\right)}_{\text{indep. of } \theta,\vartheta \ ; \ \text{depends on } \mu \ \Rightarrow\ \text{const}} \\
&= \sum_{n=1}^{N} \beta\langle \overline{A}_n^{(Y_n)},\vartheta\rangle - \ln\left(e^{\beta\langle \overline{A}_n^{(0)},\vartheta\rangle} + e^{\beta\langle \overline{A}_n^{(1)},\vartheta\rangle}\right) + \text{const} \\
\mathcal{L}_3 &= \ln P(\vartheta \,|\, \theta) + \ln P(\theta) \\
&= \frac{d}{2}\ln\left(\frac{2\pi}{\lambda^2}\right) - \frac{\lambda^2}{2}||\theta-\vartheta||_2^2 - \frac{1}{2}\ln\big(|2\pi\Sigma_0|\big) - \frac{1}{2}\theta^T\Sigma_0^{-1}\theta.
\end{aligned} \tag{6.2}$$

Hence, final surrogate loss function is

$$\begin{aligned}
\mathcal{L}(\theta,\vartheta) &= \mathcal{L}_1(\theta,\vartheta) + \mathcal{L}_2(\theta,\vartheta) + \mathcal{L}_3(\theta,\vartheta), \qquad \text{where} \\
\mathcal{L}_1(\theta,\vartheta) &= \frac{1}{2}\sum_{s=1}^{t-1}\big(R_s - \langle A_s,\theta\rangle\big)^2 \\
\mathcal{L}_2(\theta,\vartheta) &= -\sum_{n=1}^{N} \beta\langle \overline{A}_n^{(Y_n)},\vartheta\rangle + \ln\left(e^{\beta\langle \overline{A}_n^{(0)},\vartheta\rangle} + e^{\beta\langle \overline{A}_n^{(1)},\vartheta\rangle}\right) \\
\mathcal{L}_3(\theta,\vartheta) &= \frac{\lambda^2}{2}||\theta-\vartheta||_2^2 + \frac{1}{2}\theta^T\Sigma_0^{-1}\theta.
\end{aligned} \tag{6.3}$$

Finally the problem in Equation (6.1) becomes equivalent as follows:

$$(\theta_{opt},\vartheta_{opt}) = \underset{\theta,\vartheta}{\operatorname{argmax}}\ P(\theta,\vartheta \,|\, \mathcal{D}_t) \equiv \underset{\theta,\vartheta}{\operatorname{argmin}}\ \mathcal{L}(\theta,\vartheta) \tag{6.4}$$

6.2 Perturbing the loss function

Since posterior update in Equation (5.2) is intractable, we have developed the surrogate loss function above. However, it yields point estimates *only*, and hence does not give us any information about the posterior *distribution*. Hence, the idea is to *perturb* the loss function with some noise, so that the MAP

Algorithm 1 Practical Thompson Sampling

```
Input: Time horizon $T$, offline dataset $\mathcal{D}_0$, set of arms $\mathcal{A}$, knowledgeability $\lambda$ and deliberateness $\beta$.
for $t = 1, 2, \dots, T$ do
    Sample a set of perturbations $\mathcal{P}_t = \{\zeta_s, \omega_n, \theta', \vartheta'\}$.
    Solve Equation (6.5) using this set $\mathcal{P}_t$ to find $(\widehat{\theta}_t, \widehat{\vartheta}_t)$.
    Take action $A_t = \mathrm{argmax}_{a \in \mathcal{A}} \langle a, \widehat{\theta}_t \rangle$ to receive reward $R_t$.
end for
```

point estimates form a surrogate posterior distribution, and the single MAP point estimate we get serves as a *sample* drawn from that posterior, mimicking the true posterior [Osband et al., 2019, Lu and Van Roy, 2017, Qin et al., 2022].

To this extent, we propose perturbation of the 'online' loss function $\mathcal{L}_1(\cdot)$ by additive Gaussian noise, of the 'offline' loss function $\mathcal{L}_2(\cdot)$ by multiplicative random weights, and of the 'prior' loss function $\mathcal{L}_3(\cdot)$ by random samples from the prior distribution. Quantitatively, the perturbations are as follows:

- *Online perturbation.* Let $\zeta_s \sim \mathcal{N}(0, 1)$, all i.i.d. Then, the perturbed $\mathcal{L}_1(\cdot)$ becomes
$$\mathcal{L}_1'(\theta, \vartheta) = \frac{1}{2} \sum_{s=1}^{t-1} \big(R_s + \zeta_s - \langle A_s, \theta \rangle \big)^2$$
- *Offline perturbation.* Let $\omega_n \sim \exp(1)$, all i.i.d. Then, the perturbed $\mathcal{L}_2(\cdot)$ becomes
$$\mathcal{L}_2'(\theta, \vartheta) = -\sum_{n=1}^{N} \omega_n \left[\beta \langle \overline{A}_n^{(Y_n)}, \vartheta \rangle + \ln \left(e^{\beta \langle \overline{A}_n^{(0)}, \vartheta \rangle} + e^{\beta \langle \overline{A}_n^{(1)}, \vartheta \rangle} \right) \right]$$
- *Prior perturbation.* Let $\theta' \sim \mathcal{N}(\mu_0, \Sigma_0)$ and $\vartheta' \sim \mathcal{N}(\mu_0, \mathbf{I}/\lambda^2)$, all i.i.d. Then, the perturbed $\mathcal{L}_3(\cdot)$ becomes
$$\mathcal{L}_3'(\theta, \vartheta) = \frac{\lambda^2}{2} ||\theta - \vartheta - (\theta' - \vartheta')||_2^2 + \frac{1}{2} (\theta - \theta')^T \Sigma_0^{-1} (\theta - \theta')$$

So then, at each time t, let
$$(\widehat{\theta}_t, \widehat{\vartheta}_t) = \underset{\theta, \vartheta}{\mathrm{argmin}}\ \mathcal{L}'(\theta, \vartheta) = \underset{\theta, \vartheta}{\mathrm{argmin}}\ \mathcal{L}_1'(\theta, \vartheta) + \mathcal{L}_2'(\theta, \vartheta) + \mathcal{L}_3'(\theta, \vartheta). \tag{6.5}$$

The final practical algorithm then can be described in Algorithm 1.

6.3 Detailed Pseudocode

1. **Fixing the hyperparameters**.
   (a) Fix the size N of offline dataset, horizon T for the online phase, the number of arms K, dimension d of each action.
   (b) Fix μ_0 and Σ_0 so that we can have initial prior of θ as $\nu_0(\theta) \sim \mathcal{N}(\mu_0, \Sigma_0)$. Draw a sample $\widehat{\theta}_0 \sim \nu_0(\theta)$ and provide to the expert, who has a fixed knowledgeability λ and deliberateness β. The expert's knowledge then follows $\mathcal{N}(\widehat{\theta}_0, \mathbf{I}_d/\lambda^2)$, from which the expert samples a vector $\widehat{\vartheta}_0$.
2. **Generate offline dataset** $\mathcal{D}_0$.
   (a) Construct N pairs of (uniformly) sampled actions, and prefer one action over the other by following Equation (2.1) to construct the offline dataset.
3. **Begin online phase**. For each time $t = 1, 2, \dots, T$
   (a) Generate perturbation set $\mathcal{P}_t$ according to Section 6.2.
   (b) Using a convex solver solve Equation (6.5) to get $(\widehat{\theta}_t, \widehat{\vartheta}_t)$.
   (c) Take action $A_t = \mathrm{argmax}_{a \in \mathcal{A}} \langle a, \widehat{\theta}_t \rangle$ to receive reward R_t. Append $\{A_t, R_t\}$ to the dataset.

7 Conclusion and Future Scope

- The results are presented below. We proved that warmTS yield lesser regret up to 40% compared to regular TS.

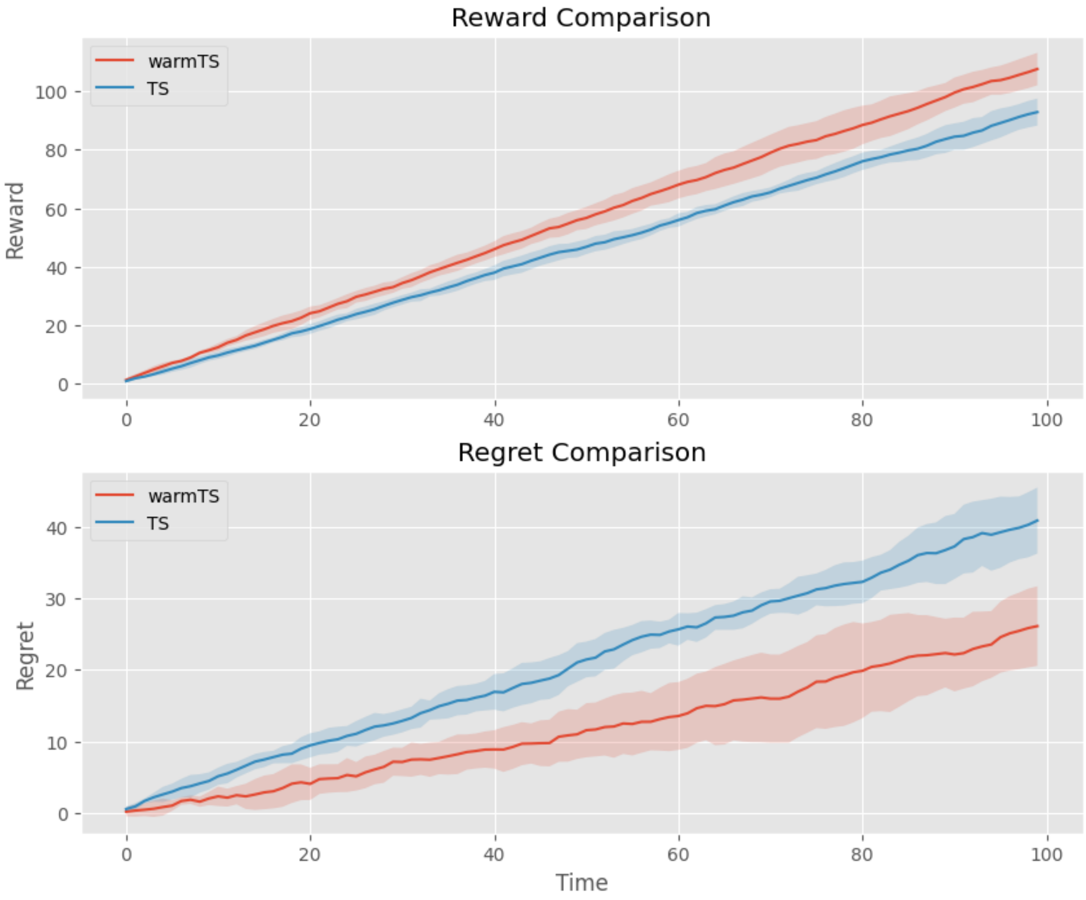

Figure 1: Reward and Regret for $N = 10$, $k = 10$, $T = 100$, $d = 5$,$\lambda = 10$,$\beta = 15$

- This project answers the question mentioned in the Introduction of how offline demonstration data can be characterized to generate portable insights.
- We can further experiment with hyper parameters λ and β to analyze its effect on reward and regret graph.
- We can also study how offline demonstration data still continues to propagate it's effect in cumulative reward with each time step.

References

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Xiuyuan Lu and Benjamin Van Roy. Ensemble sampling. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf`.

Ian Osband, Benjamin Van Roy, Daniel J Russo, Zheng Wen, et al. Deep exploration via randomized value functions. *J. Mach. Learn. Res.*, 20(124):1–62, 2019.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Chao Qin, Zheng Wen, Xiuyuan Lu, and Benjamin Van Roy. An analysis of ensemble sampling. *Advances in Neural Information Processing Systems*, 35:21602–21614, 2022.

Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34:11702–11716, 2021.

Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96, 2018.

William R Thompson. ON THE LIKELIHOOD THAT ONE UNKNOWN PROBABILITY EXCEEDS ANOTHER IN VIEW OF THE EVIDENCE OF TWO SAMPLES. *Biometrika*, 25(3-4):285–294, 12 1933. ISSN 0006-3444. doi: 10.1093/biomet/25.3-4.285. URL `https://doi.org/10.1093/biomet/25.3-4.285`.

Andrew Wagenmaker and Aldo Pacchiano. Leveraging offline data in online reinforcement learning. *arXiv preprint arXiv:2211.04974*, 2022.

Tengyang Xie, Nan Jiang, Huan Wang, Caiming Xiong, and Yu Bai. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. *Advances in neural information processing systems*, 34:27395–27407, 2021.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.