Latent Preference Bandits

Newton Mwai
mwai@chalmers.se

Emil Carlsson
emil@sleepcycle.com

Fredrik D. Johansson
frejohk@chalmers.se

Abstract

Bandit algorithms are guaranteed to solve diverse sequential decision-making problems, provided that a sufficient exploration budget is available. However, learning from scratch is often too costly for personalization tasks where a single individual faces only a small number of decision points. Latent bandits offer substantially reduced exploration times for such problems, given that the joint distribution of a latent state and the rewards of actions is known and accurate. In practice, finding such a model is non-trivial, and there may not exist a small number of latent states that explain the responses of all individuals. For example, patients with similar latent conditions may have the same preference in treatments but rate their symptoms on different scales. With this in mind, we propose relaxing the assumptions of latent bandits to require only a model of the *preference ordering* of actions in each latent state. This allows problem instances with the same latent state to vary in their reward distributions, as long as their preference orderings are equal. We give a posterior-sampling algorithm for this problem and demonstrate that its empirical performance is competitive with latent bandits that have full knowledge of the reward distribution when this is well-specified, and outperforms them when reward scales differ between instances with the same latent state.

1 Introduction

Personalized decision-making has promised to revolutionize healthcare for decades but its full impact has yet to come. In the abstract, the problem is equivalent to recommendations for media consumption or ad placement, in which algorithms make decisions and observe outcomes (reward) tailored for an individual (problem instance). Bandit and reinforcement learning algorithms are well-suited for such problems and have seen academic and commercial success (Li et al., 2010; Chapelle and Li, 2011; Bouneffouf et al., 2020; Yancey and Settles, 2020; O'Brien et al., 2022). However, the scale of data varies substantially with applications: a consumer may be exposed to dozens of recommendations in a single session, and hundreds over their subscription to a service, but a patient may try only a handful of treatments. Given their sample-hungry nature, learning a personalized treatment regime using a classical bandit algorithm to explore solutions for a single individual is usually infeasible.

A promising solution to personalization with short exploration times is to leverage structural similarities between problem instances (e.g., patients). Contextual bandits is a well-studied solution that models the expected reward of actions as fixed functions of an observed context variable (Lattimore and Szepesvári, 2020). However, this assumes that two instances would yield the same reward on average for the same action, were they in the same situation. In practice, observed contexts are rarely rich enough to capture all individual preferences. For example, in problems where rewards represent the subjective rating of an experience, different people tend to have different internal rating scales, unobserved by the learning algorithm. Latent bandits (Maillard and Mannor, 2014) attempt to overcome this limitation by allowing the reward function to depend on a latent state, observed partially and noisily through the context and the outcome of actions. If the latent state is small relative to the number of actions, or the reward is a simpler function of the latent state than the context, this structure may be sufficient to substantially reduce exploration times (Hong et al., 2020a; Kinyanjui et al., 2023). However, existing works on latent bandits assume that the full posterior distribution of the latent state is known at inference time but give little or no guidance for how it can be learned.

Contributions. 1) We propose *Latent Preference Bandits (LPB)*—a variant of latent bandits where each discrete latent state defines a preference ordering over actions, but not a full distribution of rewards. 2) We show that knowing only the set of possible preference orderings can still substantially lower the difficulty of the problem, as determined by the number of constraints added to an instance-specific lower bound on the asymptotic regret. 3) Moreover, because states only determine preference orderings, we show that LPB accommodates generalization between problem instances (e.g., patients) with different absolute reward scales, but with shared relative preferences. 4) We also show that knowledge of the set of possible preferences gives a bound on the posterior probability of the latent state and leverage this result in a regret minimization algorithm **lpbTS**, based on sampling from the approximate posterior. 5) We show empirically that it is comparable to latent bandits with fully known reward distributions when instance rewards lie in the same scale, and outperform them when instances differ in absolute reward scales. 6) Our experiments confirm that the benefit of utilizing preference structure over non-latent baselines increases as the number of arms (preference constraints) grows much larger than the number of states.

2 Related work

The latent bandit model was first studied by Maillard and Mannor (2014) in the regret minimization setting and has later been revisited for regret minimization by Atan et al. (2018); Hong et al. (2020a); Pal et al. (2023); Balcıoğlu et al. (2024) and for best-arm identification by Kinyanjui et al. (2023). Recent works have also extended the latent bandit to a non-stationary version where the latent variable evolve over time (Hong et al., 2020b; Nelson et al., 2022; Russo et al., 2024). A common theme in these works is that the reward distributions are determined completely by the latent state. This differs from our setting in the sense that we assume the latent state only defines an *ordering* of the arms.

Bandit problems with preference feedback have been widely studied in dueling bandits (Yue and Joachims, 2009; Sui et al., 2018; Bengs et al., 2021; Bergström et al., 2024) which are a bandit class where action pairs are selected at each round, and the rewards are independent, stochastic preference feedback of which arm is preferred (Sui et al., 2018). The goal is to identify the best arm, or minimize regret via pairwise comparisons (Ailon et al., 2014; Bengs et al., 2021). In dueling bandits, preferences are typically directly observed (Bengs et al., 2021), and often modeled with either utility functions (Yue and Joachims, 2009) or Probabilistic models like Bradley-Terry Models (BTMs) (Bradley and Terry, 1952; Vigneau et al., 1999). Our setting differs from dueling bandits in that we observe feedback as absolute numerical rewards for a single action, and preferences are inferred from these, in contrast to being observed directly through relative preference feedback.

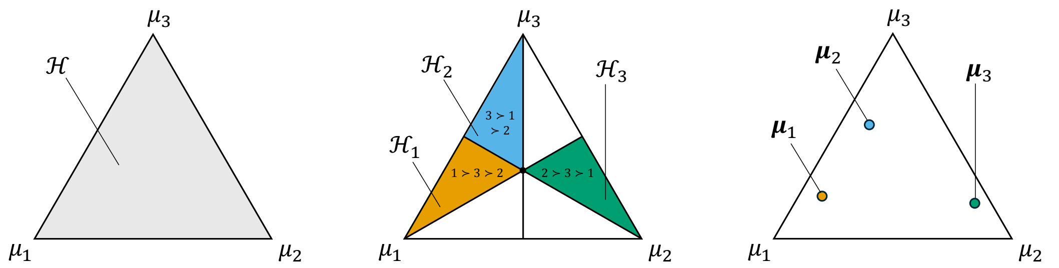
Contextual bandits (Chu et al., 2011; Agrawal and Goyal, 2013; Zhou, 2015; Lattimore and Szepesvári, 2020) exploit structure between context, actions, and rewards, promising high personalisation. However, they often assume a fixed expected reward across instances of the same context-action pair, which is inapplicable whenever no such context variable exists, even when the action preference is maintained. In our current work, we do not consider rewards structured based on context variables, but focus on the utility of latent preferences instead.

3 Multi-armed and latent bandits

We study sequential decision-making problems where the goal is to select actions $a \in \mathcal{A} = \{1, ..., k\}$ to maximize the corresponding reward R_a, *regret minimization* problems. The goal is defined with respect to the unknown expectations of rewards $\mu_a \coloneqq \mathbb{E}[R_a]$, with the optimal action $a^* = \arg\max_a \mu_a$ and optimal reward $\mu^* = \mu_{a^*}$. We aim to select actions a_t according to a policy π on times steps $t = 1, ..., T$ until a horizon T to accumulate as little regret $\mathrm{Reg}(T)$ as possible,

$$\underset{\pi}{\text{minimize}} \;\; \mathrm{Reg}(T) \quad \text{with} \quad \mathrm{Reg}(T) \coloneqq \sum_{t=1}^{T} \mathbb{E}_\pi[\mu^* - R_{A_t}] \, . \tag{1}$$

A central challenge in multi-armed bandits (MAB) is that solving (1) requires excessively many trials, especially when the number of arms, k is large. To remedy this, *latent bandits* (Maillard and Mannor,



(a) Multi-armed bandit. No constraints on reward means

(b) Latent preference bandit (this work). Known possible orderings, unknown reward means

(c) Latent bandit: Fully known possible reward means

Figure 1: Illustration of the latent preference bandit and related problems for reward means on the 2-simplex $\boldsymbol{\mu} \in \Delta^{k-1}$. In the MAB problem, no structure is known. In latent bandits, the full vector of reward means μ_z is known for each latent state z. In latent preference bandits, only the set of possible orderings is known (shown as colored segments) but two problem instances with the same latent state z may differ in their means as long as the orderings of their reward means are equal.

2014), categorize instances based on a discrete *latent* variable $Z \in \{1, ..., m\}$, representing, for example, the disease subtypes of patients with a given condition. In this way, one instance can inform another one with the same latent state. If the conditional distribution of rewards $p(R_a \mid Z = z)$ and marginal distribution $p(Z = z)$ for each latent state z are known or can be learned, these are sufficient to *infer* Z for a new instance and to minimize cumulative regret (Hong et al., 2020a).

Hong et al. (2020a) proposed the latent bandit algorithm **mTS** for regret minimization based on Thompson sampling (Thompson, 1933b; Agrawal and Goyal, 2012). At each round t, the algorithm samples a state z_t from the posterior distribution (ignoring context variables here)

$$p(Z = z \mid a_1, ..., a_{t-1}, r_1, ..., r_{t-1}) \propto \prod_{s=1}^{t-1} p(R_{a_s} = r_s \mid z) p(Z = z)$$

and selects the highest-reward action of that state, $a_z^* = \arg\max_a \mu_{a,z}$ where $\mu_{a,z} := \mathbb{E}[R_a \mid Z = z]$.

Estimating a full latent-variable model $\mathcal{M} = (p(Z), \{p(R_a \mid Z)\}_{a=1}^k)$ is nontrivial. It may not be identifiable from observable information and may require a very large data set even if it is. Moreover, requiring that all instances with state $Z = z$ follow the same reward distribution $p(R_a \mid Z = z)$ prevents instances from having individual *reward scales*, as in the example of subjective ratings from the Introduction. Fortunately, as we will see, knowing the complete likelihood is not necessary to achieve benefits over learning tabula rasa for a new bandit instance. In fact, knowing only the optimal arm in each latent state leads to improved regret bounds when $m < k$. However, imposing additional structure on the rewards of different actions can help distinguish the true latent state from alternatives. To this end, we study latent bandits with states defined by *preference orderings* of actions.

4 Latent preference bandits

We introduce the *latent preference bandit* (LPB) problem and give an instance-dependent lower bound for the best achievable regret which depends on the structured preferences in rewards. Next, we propose a posterior-sampling algorithm to solve the problem and discuss its performance.

In latent preference bandits, a problem instance $(z, \mathcal{P})$ is defined by the latent state z and reward distributions for the k actions, $\mathcal{P} = (P_1, ..., P_k)$ structured according to z. For clarity, we focus on Gaussian rewards with equal variance σ^2 where $P_a = \mathcal{N}(\mu_a, \sigma^2)$ and, for the remainder of the paper, we represent instances $(z, \boldsymbol{\mu})$ by their latent states and reward means $\boldsymbol{\mu} = [\mu_1, ..., \mu_k]^\top$. Each latent state $z \in [m]$ is associated with a preference ordering of actions $O_z = (o_{z,1}, ..., o_{z,k})$ such that the expected rewards for problem instances $(z, \boldsymbol{\mu})$ are ordered according to O_z, i.e.,

$\mu_{o_{z,1}} \geq \mu_{o_{z,2}} \geq \cdots \geq \mu_{o_k}$.[1] For convenience, let $i_{a,z}$ denote the rank of action a under O_z, that is $i_{a,z} = j$ such that $o_{z,j} = a$, and let $a \succeq_z a' \Leftrightarrow i_{a,z} < i_{a',z}$ denote that a is preferred to a' under z.

Two problem instances $(z, \boldsymbol{\mu}), (z', \boldsymbol{\mu}')$ with the same latent state $z = z'$ are guaranteed to have the same preference orderings but may not have the same distributions of rewards. This allows for modeling individual rating scales, as in the following example: Two patients with a chronic condition share the same subtype of disease z which determines what therapies $a \in \mathcal{A}$ are preferred over which other therapies. However, the two patients have different tolerance for pain and give different ratings R_a for their symptoms under treatment a even if their relative preferences are the same. We let $\mathcal{H}_z = \{\boldsymbol{\mu} \in \mathbb{R}^k : \forall a \succeq_z a' : \mu_a \geq \mu_{a'}\}$ denote the set of all reward parameters consistent with z.

At the start of learning, algorithms are assumed to have access to a model of $\{(z, O_z)\}_{z \in \mathcal{Z}}$ of preference orderings but have no knowledge of rewards—they only know which preference orderings are possible. We assume that all latent states have unique orderings. We illustrate the LPB problem in Figure 1 for the special case of reward means in the 2-dimensional simplex, and compare it to the standard multi-armed bandit (MAB) and the latent bandit problem from Hong et al. (2020a). The LPB problem has more structure than MAB but less than full-model latent bandits. To see this, we can adapt an instance-specific lower bound on the asymptotic regret from Maillard and Mannor (2014), in turn adapted from Agrawal et al. (1989).

Theorem 1 (Agrawal et al. (1989))**.** *Let $(z, \boldsymbol{\mu})$ be the true latent state and reward means of an instance and let $\mathcal{A}_- = \mathcal{A} \setminus \{a^*_z\}$ be the set of sub-optimal arms a with optimality gaps $\Delta_a = \mu^* - \mu_a$. Then, for any uniformly good control scheme, i.e., that achieves $\mathrm{Reg}(T) = o(T^b)$ for any $b > 0$,*

$$\liminf_{T \to \infty} \frac{\mathrm{Reg}(T)}{\log T} \geq \min_{w_c \in \mathcal{P}(\mathcal{A}_-)} \max_{\lambda \in \mathrm{Alt}(z, \boldsymbol{\mu})} \frac{\sum_{a \in \mathcal{A}_-} w_a \Delta_a}{\sum_{a \in \mathcal{A}_-} w_a \mathbb{KL}(\mu_a || \lambda_a)}, \qquad (2)$$

where $\mathrm{Alt}(z, \boldsymbol{\mu}) = \cup_{z : a^*_{z'} \neq a^*_z} \mathcal{H}_z$, $\mathcal{P}(\mathcal{A}_-)$ is the simplex over $\mathcal{A}_-$, and $\mathbb{KL}(\mu_a || \lambda_a)$ is the KL-divergence between Gaussian distributions with unit variance and means μ_a, λ_a.

The set $\mathrm{Alt}(z, \boldsymbol{\mu})$ of alternative instances contain only parameters μ' consistent with orderings that are different from that of the true state, O_z. If every possible ordering is represented by a latent state ($m = k!$), the problem is difficult, comparable to an MAB without structure. On the other hand, if $m \ll k!$ and the orderings of latent states are random, the closest confusing instance λ^*, maximizing (2), is likely to have an ordering with a large number of inversions to $\boldsymbol{\mu}$, the $\mathbb{KL}$ term will be large, and the bound small—the problem is easier to solve. We return to this discussion in Section 4.2.1.

4.1 Regret minimization with absolute feedback

We focus on learning from absolute feedback[2] where decision-makers select a single action $a_t \in [k]$ at each round t and observe a stochastic reward $R_t \in \mathbb{R}$, generated according to the unknown instance $(z, \boldsymbol{\mu})$ with $\boldsymbol{\mu} \in \mathcal{H}_z$. Disregarding the latent state, the setting coincides with the classical MAB problem. Thus, the primary target for algorithms that exploit knowledge of the set of possible latent states $z \in [m]$ and their preference orderings O_z is to minimize regret or identify the optimal arm more rapidly than MAB algorithms and other algorithms that exploit the latent structure.

The worst-case asymptotic regret for the MAB problem is well-known to be $\Theta(\sqrt{kT})$ as $T \to \infty$, achieved by several algorithms, including upper-confidence bound (UCB) maximization and posterior (Thompson) sampling (Lattimore and Szepesvári, 2020). Hong et al. (2020a) proposed the **mTS** and **mUCB** algorithms for the latent bandit problem with m states and showed that their worst-case asymptotic regret with full knowledge of the posterior of observations, including the reward distribution, is $O(\sqrt{mT \log T})$. This matches the MAB result up to log factors, but with m instead of k, which can be very beneficial if $m < k$. However, with knowledge of the set of possible latent states, this can be achieved simply by restricting the action set.

Proposition 1. *Consider the following algorithm. Whenever $k < m$, restrict the action to the subset $\mathcal{A}^*_{\mathcal{Z}}$ of optimal $u < m$ arms of which each is optimal in at least one latent state, $\mathcal{A}^*_{\mathcal{Z}} = \{a \in [k] : \exists z \in \mathcal{Z} : a^*_z = a\}$, and run a standard MAB algorithm restricted to $\mathcal{A}^*_{\mathcal{Z}}$. When $k \geq m$, run a standard MAB algorithm on $\mathcal{A} = [k]$. This procedure achieves $O(\sqrt{\min(k, m)T})$ regret in the worst case on the latent bandit and latent preference bandit problems.*

[1] We drop the z-subscript in $\mu_{o_{z,i}}$ for readability when clear from context.

[2] In the Appendix, we briefly discuss latent preference bandits with relative (dueling) feedback.

Algorithm 1 Thompson sampling for LPB regret minimization with absolute rewards (**lpbTS**)

```
Let $\hat{p}(z) = \frac{1}{m}$ for latent states $z = 1, ..., m$
for $t = 1, ...$ do
    Sample $z_t \sim \hat{p}(z)$
    Perform action $a_t = a^*_{z_t}$
    Observe $r_t$ from the environment
    for $z = 1, ..., m$ do
        Update mean parameters $\hat{\mu}_z$ according to (3)
    end for
    Update latent state posterior $\hat{p}(Z)$ according to (4)
end for
```

Proof. The procedure uses $\min(|\mathcal{A}|, |\mathcal{A}^*_{\mathcal{Z}}|) \leq \min(k, m)$ arms, and the optimal arm for any latent bandit or LPB instance is contained in either action set. We can directly apply standard regret bounds for MAB algorithms (see e.g., Lattimore and Szepesvári (2020)) to the restricted action sets. □

Notably, the procedure in Proposition 1 does not use knowledge of the set of possible reward *means*, only the set of possible best arms. On its face, it may seem that the **mTS** algorithm of Hong et al. (2020a) is matched by simply this simple algorithm. However, as we see empirically in Section 5, this is far from true: **mTS** achieves substantially better performance by exploiting reward structure, even though this is not yet explained by theory. In fact, when all reward means are distinct across states and known, $\mu_{a,z} \neq \mu_{a,z'}$, regret constant in T is achievable for latent bandits. However, this is not achievable in the LPB setting since the mean parameters must be estimated during exploration. In short, to reach optimal empirical performance on the latent preference bandit problem, and hope to match the lower bound in (2), it is critical to exploit the structure between arm parameters.

4.2 A posterior-sampling LPB algorithm exploiting the ordering of means

We propose the **lpbTS** algorithm (Algorithm 1) for the LPB problem based on sampling from the posterior of the latent state and selecting the optimal arm for that state. First, let $\mathcal{D}_T = ((a_1, r_1), ..., (a_T, r_T))$ denote the history of the first T observations collected during exploration for a problem instance $(z, \boldsymbol{\mu})$. The likelihood of $\mathcal{D}_T$ under a state z with preference ordering O_z is then

$$\mathcal{L}(\mathcal{D}_T \mid Z = z) = \prod_{t=1}^{T} p(r_t \mid a_t, z) = \int_{\boldsymbol{\mu} \in \mathcal{H}_z} p(\boldsymbol{\mu} \mid z) \prod_{t=1}^{T} p(r_t \mid a_t, \boldsymbol{\mu}, z) d\boldsymbol{\mu} \; .$$

and can be used to construct the posterior probability $p(Z = z \mid \mathcal{D}_T)$, provided that a well-specified parameter prior $p(\boldsymbol{\mu} \mid z)$ is known for each latent state z. In general, the constraint $\boldsymbol{\mu} \in \mathcal{H}_z$ means that no closed-form expression exists, and computing it exactly is intractable. In principle, we could appeal to variational inference (Jordan et al., 1999) for an approximation, and if a strong parameter prior $p(\boldsymbol{\mu} \mid z)$ is available, this can offer substantially more information to the learner than the ordering $o_{1,z} \succeq o_{2,z} \succeq ... \succeq o_{k,z}$ implied by z. In the extreme case that $p(\boldsymbol{\mu} \mid z)$ is a delta function, this coincides with the latent bandit problem of Hong et al. (2020a). As we try to minimize the information needed about the latent variable, *we assume that no parameter prior is available*.

Without a parameter prior, the likelihood $p(r_t \mid a_t, z)$ is not fully defined, but we can construct an upper bound on the likelihood by considering the mean configuration with the highest likelihood for the data restricted to the available orderings implied by $\mathcal{Z}$. For all states z,

$$\mathcal{L}(\mathcal{D}_T \mid Z = z) = \int_{\boldsymbol{\mu} \in \mathcal{H}_z} p(\boldsymbol{\mu} \mid z) \prod_{t=1}^{T} p(r_t \mid a_t, \boldsymbol{\mu}, z) d\boldsymbol{\mu} \leq \sup_{\boldsymbol{\mu} \in \mathcal{H}_z} \prod_{t=1}^{T} p(r_t \mid a_t, \mu_{a_t}) \; .$$

With Gaussian rewards, maximizing this *upper* bound corresponds to minimizing the mean squared error of $\boldsymbol{\mu}$ in predicting the observed reward, constrained to the set $\mathcal{H}_z$. Thus, under the assumption that z is the correct latent state, we may estimate the mean parameters as follows.

$$\hat{\boldsymbol{\mu}}_z \coloneqq \operatorname*{arg\,min}_{\boldsymbol{\mu} \in \mathcal{H}_z} -\ell(\mathcal{D}_T \mid \boldsymbol{\mu}), \quad \text{where} \quad -\ell(\mathcal{D}_T \mid \boldsymbol{\mu}) \propto \sum_{t=1}^{T} \frac{(r_t - \mu_{a_t})^2}{\sigma^2} \tag{3}$$

With $\{\hat{\boldsymbol{\mu}}_z\}$ the minimizers of (3) for all z, we can construct an *optimistic* posterior estimate,

$$\forall z : \hat{p}(z \mid \mathcal{D}_t) := \frac{1}{\alpha} p(\mathcal{D}_t \mid \hat{\boldsymbol{\mu}}_z) \ . \tag{4}$$

where α is the normalization constant. Note that this is an approximation as the bound $p(\mathcal{D}_t \mid \hat{\boldsymbol{\mu}}_z) \geq p(\mathcal{D}_t \mid z)$ affects also the normalization constant of the posteriors: The looser the bound for one state, the lower the posterior probability of other states.

We design our method, **lpbTS** (Algorithm 1) for regret minimization by selecting the optimal arm for a state sampled from the approximate posterior (4). The constrained maximum-likelihood estimation (MLE) problem in (3) solved for each state is a quadratic program with linear inequality constraints. We show below that it can be solved using off-the-shelf solvers for isotonic regression (Barlow and Brunk, 1972). We discuss the computational complexity of **lpbTS** in the Appendix.

Proposition 2. *Let $n_a = \sum_{t=1}^{T} \mathbb{1}[a_t = a]$ and define $w_a = \frac{n_a}{\sigma_a^2}$. Next, let $O_z = (o_1, ..., o_k)$ be the preference ordering of latent state z. Then, the solution to the isotonic regression problem with outcomes $y_a = \frac{1}{n_a}\sum_{t:a_t=a} r_t$ and sample weights w_a*

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^d}{\textit{minimize}} \quad \sum_{a=1}^{k} w_a(\mu_a - y_a)^2 \quad \textit{subject to} \quad \mu_{o_k} \leq \mu_{o_{k-1}} \leq ... \leq \mu_{o_1}$$

solves the constrained MLE problem in (3). A proof is given in Appendix C.

4.2.1 On the scaling of regret with latent states, actions and constraints

In Section 4.1, we gave a simple procedure that achieves $O(\sqrt{\min(k,m)T})$ regret on the latent preference bandit problem by restricting the action set to the $\leq m$ arms that are optimal in at least one latent state. Although we don't prove this formally, we should expect a similar regret bound to be provable for **lpbTS**, by appealing to general results (Geyer, 1994) which state that constrained MLE has asymptotic variance that is no worse than unconstrained MLE (as used in many MAB algorithms), provided that the true parameter lies in the constraint set, which it does here. Indeed, in our experiments, both **lpbTS** and **mTS** (Hong et al., 2020a) vastly outperform an uninformed MAB algorithm, even when restricting the number of arms as described above. This gap is not predicted by worst-case upper bounds for **mTS** and MAB algorithms.

The empirical performance of **lpbTS** is affected by the nature and number of order constraints. For k arms, there are $k!$ possible orderings. Thus, even when m grows linearly with k, the space of possible parameter vectors is reduced by an $O(k!)$ factor compared to an unconstrained problem. The effect of this is largest when many of the constraints in (3) are active for states z' that are not the ground-truth state z^*, and their MLE estimate projects the empirical reward means onto $\mathcal{H}_{z'}$. The more active constraints, the larger the projection and the smaller the likelihood of observed rewards under z'. As a toy example, assuming that orderings O_z are randomly selected from all possible permutations of the k actions without replacement, the probability that two such orderings differ in only two positions is $\binom{k}{2}/(k!-1)$ (see Appendix F). For large k, this probability is vanishingly small. For example, with $k = 10$, this probability evaluates to approximately $\frac{45}{3{,}628{,}799} \approx 1.24 \times 10^{-5}$. Given that m is typically much smaller than $k!$, the ground-truth state will stand out even more as k grows. This explains why subsampling the action set as in Section 4.1 may achieve surprisingly low worst-case regret but performs substantially worse empirically than algorithms exploiting latent structure.

5 Experiments

We compare our algorithm **lpbTS**, on both synthetic and realistic tasks, to: i) standard MAB **Thompson Sampling (TS)** (Thompson, 1933a; Russo et al., 2018) that is oblivious to the latent state structure, initialized with Gaussian priors and using the ground-truth variance of the reward, ii) **TS, top arm subset** (see Proposition 1), and iii) **mTS** (Hong et al., 2020a), Thompson Sampling with a perfect latent variable model, including the exact reward mean vectors for all arms and latent states.

5.1 Synthetic Experiments

We first consider a synthetic bandit environment designed to simulate a well-specified LPB setting. The environment is parameterized by fixable variable numbers of k arms and m latent states. For

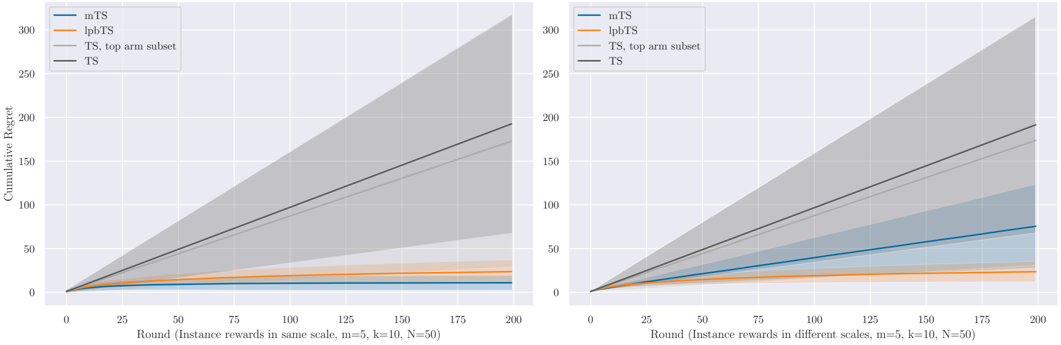


Figure 2: Synthetic experiment, cumulative regret compared to baselines ($m = 5, K = 10, N = 50$). **lpbTS (Ours)** is comparable to latent bandit baselines when instance rewards lie in the same scale (***Left***) and outperforms baselines when instance rewards in different reward scales (***Right***)

.

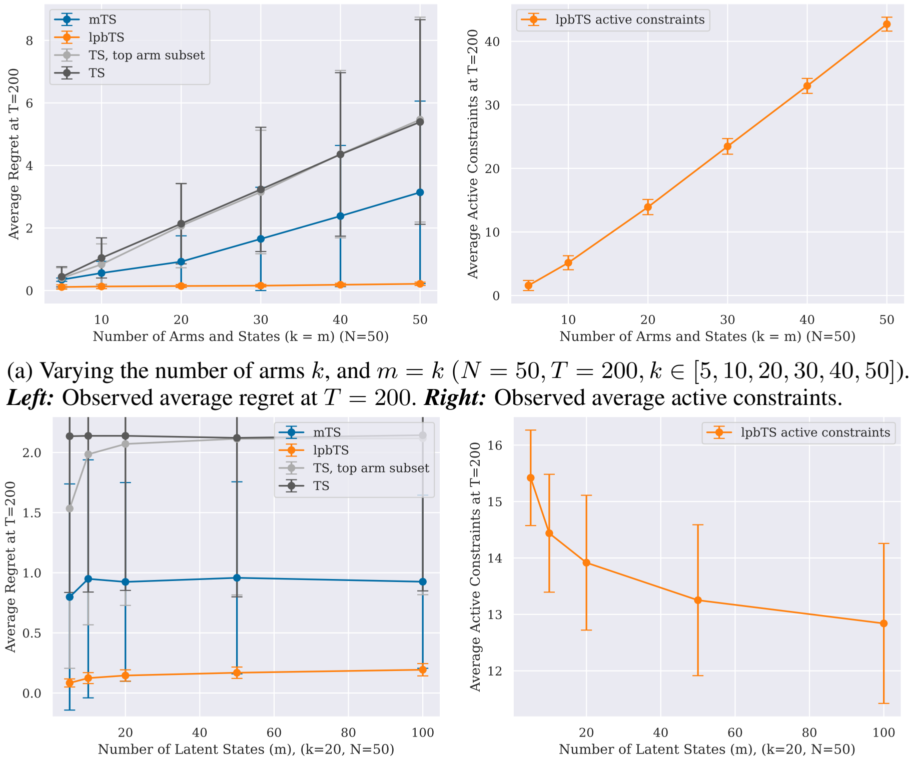
given k and m, for each latent state $z \in [m]$, a random preference ordering (permutation) $O_z = (o_{z,1}, ..., o_{z,k})$ of the arms is generated without replacement. Rewards are Gaussian distributed, with variance $\sigma^2 = 1$. The mean reward for arm a in state z, $\mu_{a,z}$ is based on its position in O_z, ensuring a strictly decreasing sequence of means along the permutation, with the optimal arm in each state, $o_{z,1}$ assigned mean $\mu_{o_{z,1},z} \sim \mathcal{U}(C + \Delta k, C + \Delta k + 1)$. We use $C = 9$, and $\Delta = 0.2$. For subsequent arms $o_{z,j}$, $j \in \{2, \ldots, k\}$, the mean is defined recursively as $\mu_{o_{z,j},z} = \mu_{o_{z,j-1},z} - \Delta - U$, where $U \sim \mathcal{U}(0, \epsilon)$ with $\epsilon = 0.05$. We also allow for reward scales to vary among instances in the same latent state, by having a larger draw interval for the mean reward of the optimal arm: $\bar{\mu}_{o_{z,1},z} \sim \mathcal{U}(\bar{C} + \Delta k,\ \bar{C} + \Delta k + \gamma k)$ with $\bar{C} = 6$ and $\gamma = 0.4$, and subsequent arm means defined recursively as before. We conduct experiments with $N = 50$ independent samples of instance means per latent state, each having $T = 200$ rounds. We report all the configurations of k, m, T and N in the results presented. We use cumulative regret over T, and average regret at final round T as the error metrics. We report these as averages over $m \times N$ independent instances, and their corresponding standard deviation as errors. We also report the average active constraints in **lpbTS**, reported at the final round T for each value of the varied parameter, with standard deviation as the error.

5.2 MovieLens Experiments

For a realistic personalisation setting, we construct a task based on the MovieLens (Harper and Konstan, 2015) datasets where actions represent movie choices, rewards are ratings of movies, and latent states are groups of users. See Appendix B for a full description. First, users are clustered into latent states based on their real-world sparse movie preference ratings. Then, preference orderings for all clusters are learned using Bradley-Terry Models (BTMs) (Bradley and Terry, 1952) and used to define reward models assigning an average rating in $\mu_{a,z} \in [1, 5]$ to each movie. Personalized rating scales for each instance are created by drawing a random rating interval defined by the smallest allowable interval length, $\zeta = 1.5$. For each experiment, we sample 100 random users and do 200 rounds of movie ratings per user, where at each round, 300 genre-diverse movies are sampled from the set of available movies to form the active action set. We evaluate how well **lpbTS** performs compared to baselines i) when movie ratings for users of the same latent state are in the same absolute scale, ii) when individual ratings could vary in scale for users, and iii) when using a *learned* latent preference ordering model $\{(z, \hat{O}_z)\}$ (see Appendix B) rather than the ground-truth model, to study the effects of possible errors in model fit and latent state recovery. Cumulative regret results are reported as averages and standard deviations over user instances. Additionally, we report the average of movie ratings achieved by the algorithms, standardized with z-score standardization *per user* to ensure consistency when rewards can vary in scale, together with standard errors.

5.3 Results

LPB structure benefits exploration, and is robust In Figure 2 (*Left*), we see that **lpbTS** is comparable in performance to **mTS**, in the comparable convergence of the cumulative regret, when instance reward means have a fixed reward scale in the latent states. However, **mTS** outperforms



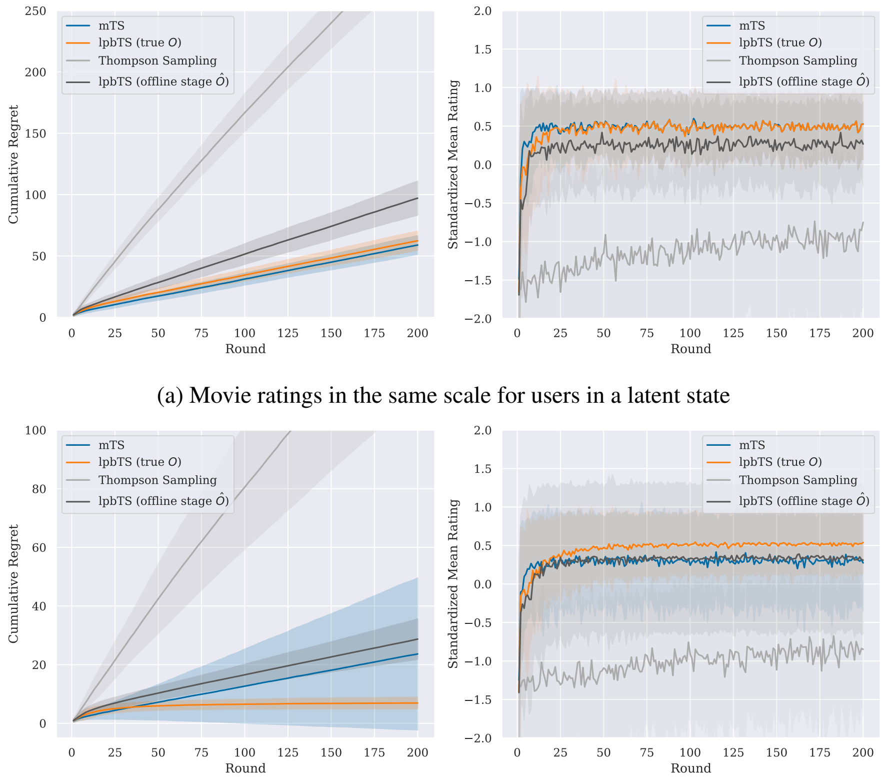
(a) Varying the number of arms k, and $m = k$ ($N = 50, T = 200, k \in [5, 10, 20, 30, 40, 50]$). ***Left:*** Observed average regret at $T = 200$. ***Right:*** Observed average active constraints.

(b) Varying the number of latent states m ($k = 20, N = 50, T = 200, m \in [5, 10, 20, 50, 100]$). ***Left:*** Observed average regret at $T = 200$ ***Right:*** Observed average active constraints at $T = 200$.

Figure 3: Synthetic experiment, instance rewards in different scales. When $m = k$ and k grows, the true latent state stands out more as its closest neighbors become more different with high probability. When m increases with k fixed, it is possible to assign more distinct means and orderings, reducing the structural benefit of latent bandits (**lpbTS**, **mTS**) over unconstrained methods (TS).

lpbTS as it converges quicker, with a lower variance because of it's knowledge of the *true* latent reward means. **lpbTS** has to *estimate* these latent means with noisy rewards. Adding the ordering constraints O is vastly beneficial compared to no stucture as in TS. Unsurprisingly, TS, top arm subset is also poor, because it does not actually exploit structure despite achieving the upper bound. In Figure 2 (*Right*), we see the benefit of using a more general latent structure O compared to latent mean vector of rewards: Cumulative regret for **lpbTS** converges quicker than that for **mTS** with a visibly lower variance. This is because the latent model comprising mean vectors is misspecified when absolute reward scales can vary for different latent state instances.

LPB characteristics become explainable from observed active constraints In Figure 3a, we observe that as k and $m = k$ increase, the average regret for **mTS** also increases while that of **lpbTS** is fairly constant. This is because instance rewards have different scales, and the draw interval for the mean reward of the optimal arm grows with Δk. If the interval was fixed, **mTS** would still be worse than **lpbTS**, but would not grow this way (e.g. see Figure 3b). We also see that the number of active constraints grow. This is because $m = O(k)$, but the number of possible permutations grow like $k!$, so the probability of having large differences between states grows when $m = k$ and k grows. This is not predicted by a $O(\sqrt{\min(k, m)T})$ bound since $m = k$. It is explained by the fact that the true latent state stands out more with high probability, and the empirical isotonic means $\hat{\mu}_z$ become less likely to align with the neighbor states (the most confusable states) relative to the true state, resulting in a higher number of active constraints. In Figure 3b, we see that when the number of latent states is increased with k fixed, the average regret increases in **mTS**, **lpbTS**, and TS, top arm subset until $m = k$ at which point it flattens. This can be explained by the worst-case $O(\sqrt{\min(k, m)T})$ bound. We defer additional empirical results and discussions aligning with these



(a) Movie ratings in the same scale for users in a latent state

(b) Movie ratings in different scales for users in a latent state

Figure 4: MovieLens Experiment, 20M Dataset. Results match theory: **lpbTS (Ours)** is comparable to mTS in (a), outperforms in (b), and the two-stage recovery of O is empirically validated.

insights to Appendix I, in Figures 6a-6c for varying Δ, k(with m fixed), and σ, and Figures 7b-**??** for instance rewards in the same scale.

MovieLens results are aligned Results from MovieLens experiments are consistent with the theory and synthetic results, where **lpbTS** is comparable (Figure 4a, 20M) to **mTS** (which uses the oracle ground truth model) when ratings are in the same scale for the latent states, and **mTS** is worse (Figure 4b, 20M) when ratings are in different scales. Further, **lpbTS** with an offline recovered model is competitive compared to **lpbTS**, oracle ground-truth model, albeit biased especially when user ratings are in different scales. We re-iterate that recovering latent models is a key ingredient for latent bandits, and we empirically demonstrate a method to recover them for Latent Preference Bandits with our two-stage recovery, with many actions. Results for 1M and 32M datasets are left to Appendix I.

6 Conclusion

We have proposed Latent Preference Bandits (LPB), a novel bandit problem where instances share structure based on a latent variable that determines the preference ordering of actions but not their rewards. We design the algorithm **lpbTS** for regret minimization in this setting by sampling from an approximate posterior of the latent state, constrained by the set of possible orderings. We demonstrate that, despite using less information, **lpbTS** is competitive with latent bandit algorithms that have full knowledge of the reward distribution of each arm when all instances of the same state have the same distribution, and outperforms them when individual rating scales differ between instances who share the same preference ordering. The benefit over uninformed bandit algorithms grows when the number of latent states (orderings) is small relative to the number of arms. A limitation of our algorithm is that it requires a model of the preference ordering of all latent states. However, in our experiments on movie ratings, we find that a learned model performs comparably to the ground truth.

References

Rajeev Agrawal, Demosthenis Teneketzis, and Venkatachalam Anantharam. Asymptotically efficient adaptive allocation schemes for controlled i. i. d. processes: Finite parameter space. *IEEE Transactions on Automatic Control*, 34(3):258–267, 1989.

Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, pages 39–1. JMLR Workshop and Conference Proceedings, 2012.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International conference on machine learning*, pages 127–135. PMLR, 2013.

Nir Ailon, Zohar Karnin, and Thorsten Joachims. Reducing dueling bandits to cardinal bandits. In *International Conference on Machine Learning*, pages 856–864. PMLR, 2014.

Onur Atan, Cem Tekin, and Mihaela van der Schaar. Global bandits. *IEEE transactions on neural networks and learning systems*, 29(12):5798–5811, 2018.

Ahmet Zahid Balcıoğlu, Emil Carlsson, and Fredrik D Johansson. Identifiable latent bandits: Combining observational data and exploration for personalized healthcare. *arXiv preprint arXiv:2407.16239*, 2024.

Richard E Barlow and Hugh D Brunk. The isotonic regression problem and its dual. *Journal of the American Statistical Association*, 67(337):140–147, 1972.

Viktor Bengs, Róbert Busa-Fekete, Adil El Mesaoudi-Paul, and Eyke Hüllermeier. Preference-based online learning with dueling bandits: A survey. *Journal of Machine Learning Research*, 22(7): 1–108, 2021.

Herman Bergström, Emil Carlsson, Devdatt Dubhashi, and Fredrik D. Johansson. Active preference learning for ordering items in- and out-of-sample. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=PSLH5q7PFo.

Djallel Bouneffouf, Irina Rish, and Charu Aggarwal. Survey on applications of multi-armed and contextual bandits. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2020. doi: 10.1109/CEC48606.2020.9185782.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. *Advances in neural information processing systems*, 24, 2011.

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214. JMLR Workshop and Conference Proceedings, 2011.

Charles J Geyer. On the asymptotics of constrained m-estimation. *The Annals of statistics*, pages 1993–2010, 1994.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

Joey Hong, Branislav Kveton, Manzil Zaheer, Yinlam Chow, Amr Ahmed, and Craig Boutilier. Latent bandits revisited. *Advances in Neural Information Processing Systems*, 33:13423–13433, 2020a.

Joey Hong, Branislav Kveton, Manzil Zaheer, Yinlam Chow, Amr Ahmed, Mohammad Ghavamzadeh, and Craig Boutilier. Non-stationary latent bandits. *arXiv preprint arXiv:2012.00386*, 2020b.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37:183–233, 1999.

Newton Mwai Kinyanjui, Emil Carlsson, and Fredrik D. Johansson. Fast treatment personalization with latent bandits in fixed-confidence pure exploration. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=NNRIGE8bvF. Expert Certification.

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

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

Odalric-Ambrym Maillard and Shie Mannor. Latent bandits. In *International Conference on Machine Learning*, pages 136–144. PMLR, 2014.

Elliot Nelson, Debarun Bhattacharjya, Tian Gao, Miao Liu, Djallel Bouneffouf, and Pascal Poupart. Linearizing contextual bandits with latent state dynamics. In *Uncertainty in Artificial Intelligence*, pages 1477–1487. PMLR, 2022.

Conor O'Brien, Huasen Wu, Shaodan Zhai, Dalin Guo, Wenzhe Shi, and Jonathan J Hunt. Should i send this notification? optimizing push notifications decision making by modeling the future. *arXiv preprint arXiv:2202.08812*, 2022.

Soumyabrata Pal, Arun Sai Suggala, Karthikeyan Shanmugam, and Prateek Jain. Optimal algorithms for latent bandits with cluster structure. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 7540–7577. PMLR, 25–27 Apr 2023.

Alessio Russo, Alberto Maria Metelli, and Marcello Restelli. Switching latent bandits. *Transactions on Machine Learning Research*, 2024.

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.

Yanan Sui, Masrour Zoghi, Katja Hofmann, and Yisong Yue. Advancements in dueling bandits. In *IJCAI*, pages 5502–5510, 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, 1933a. ISSN 00063444. URL http://www.jstor.org/stable/2332286.

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, 1933b.

E Vigneau, Ph Courcoux, and M Semenou. Analysis of ranked preference data using latent class models. *Food quality and preference*, 10(3):201–207, 1999.

Kevin P Yancey and Burr Settles. A sleeping, recovering bandit algorithm for optimizing recurring notifications. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3008–3016, 2020.

Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1201–1208, 2009.

Li Zhou. A survey on contextual multi-armed bandits. *arXiv preprint arXiv:1508.03326*, 2015.

A Notation

A list of common notations is given in Table 1. Generally, capital Roman letters denote random variables and lower-case Roman letters denote constants or observations or random variables. The sequence O_z is an exception since it is not random. Caligraphic Roman letters denote sets.

Table 1: Common notation

s, t	Time indices
T	Time horizon
$\mathcal{A}$	Set of available actions
a	A single action in $\mathcal{A}$. a_t is the action at time t
$\mathcal{Z}$	Set of latent states
Z	Latent variable on $\mathcal{Z}$
z	A single latent state in $\mathcal{Z}$
R_a	Stochastic reward for action a
r_t	Observation of reward at time t following action a_t
μ_a	Expected reward under action a, $\mu_a = \mathbb{E}[R_a]$
σ_a	Standard deviation of reward under action a, $\sigma_a^2 = \mathbb{V}[R_a]$
$\boldsymbol{\mu}$	Vector of reward means for all actions
a^*	Action with the highest expected reward
μ^*	Optimal reward, reward of optimal action
$\mu_{a,z}$	Expected reward under action a in latent state z
σ_a	Standard deviation of reward under action a in latent state z
$\boldsymbol{\mu}_z$	Vector of reward means in latent state z for all actions
$\mathcal{H}_z$	Set of valid reward means in latent state z, $\boldsymbol{\mu}_z \in \mathcal{H}_z$
$\mathcal{H}$	Set of globally valid reward means, $\boldsymbol{\mu} \in \mathcal{H} = \cup_{z \in \mathcal{Z}} \mathcal{H}_z$
a_z^*	Action with the highest expected reward in latent state z
μ_z^*	Optimal reward for any action in latent state z
$\hat{\mu}_a, \hat{\boldsymbol{\mu}}, \hat{\mu}_{a,z}, \hat{\boldsymbol{\mu}}_z$	Estimates of reward means corresponding to the above
π	Decision making policy to select a_t
$\text{Reg}(T)$	Cumulative regret until horizon T
$\mathcal{M}$	Latent-variable model
$O_z = (o_{z,1}, ..., o_{z,k})$	Preference ordering of actions for latent state z
$i_{a,z}$	Rank of action a under latent state z
ρ	Separation parameter for preference probability
Δ	Separation parameter for reward means
$\tilde{R}_t$	Preference feedback at time t
$\mathcal{D}_T$	History of observations up to time T
w_a	Weight parameter in isotonic regression

A.1 Errata

- Line 291: Typo: There's an extra "increase" - removed
- Line 299: Typo: "compared to **lpbTS** the" - "compared to **lpbTS**,"

B Details on the MovieLens experiments

For a real-world personalisation setting, we do experiments on the MovieLens (Harper and Konstan, 2015) datasets, using the 1M, 20M and 32M versions. For each dataset, we first filter out movies that have less than 200 ratings, and users who have rated less than 200 movies. The resulting sizes are *1M:* 1,589 users, 1,132 movies, 498,677 ratings, *20M:* 26,826 users 6,236 movies, 11,905,303 ratings, *32M:* 42,197 users 8,777 movies, 19,832,758 ratings. We use $k =$ *#movies* after filtering. We then obtain a train-test split on users with a split ratio of 0.5, and further split the test user set into a bandit inference set, and an *"offline estimation"* set with a split ratio of 0.5. To get the ground-truth latent preference orderings O, a two-stage preference ordering recovery proceeds as follows: i) KMeans clustering on the training set, with 5 clusters ($m = 5$) on the sparse ratings. ii) For all users assigned to a cluster z, the cluster-specific preference orderings for all movies are obtained by fitting preference orderings on the users' movie ratings, with Bradley-Terry Models (BTMs) (Bradley and Terry, 1952), see Appendix H. Our two-stage BTM-based recovery yields a set of $\beta_{z,a} \in [0, 1], z \in [m], a \in [k]$ where $\beta_{z,a}$ define the preference strength for movie a in cluster z after sigmoid transformation, and O_z is the indices of the decreasing sort of β_z. This two-stage preference ordering recovery is motivated by our observation that the recovery is comparable in performance to a BTM Expectation-Maximization (EM) approach for latent preference recovery, like the one proposed by Vigneau et al. (1999), see Appendix Figure 5. For a user at bandit time, their true z (obtained, using the KMeans model fit in the two stage recovery) is used to get the environment reward. The environment reward for a user in latent state z, and for movie k is obtained as $\mathcal{N}(\bar{\beta}_{z,k}, 0.5)$, where $\bar{\beta}_z$ is a scaling of β_z to the MovieLens rating scale [1, 5] using $\bar{\beta}_z = 1 + 4 \times \frac{\beta_z - \min(\beta_z)}{\max(\beta_z) - \min(\beta_z)}$. We also allow for instance rewards to vary in scale to $\bar{\bar{\beta}}_z$ computed as: $\bar{\bar{\beta}}_z = \eta + (\upsilon - \eta) \times \frac{\bar{\beta}_z - 1}{5 - 1}$ where η is their possible minimum rating with $\eta \sim \mathcal{U}(1, 5 - \zeta)$, and υ is their possible maximum rating with $\upsilon \sim \mathcal{U}(\eta + \zeta, 5)$ and ζ is the allowable minimum rating interval with $\zeta = 1.5$. For varying individual scales, we only consider reward randomness due to the random rating perturbation.

For each experiment, we sample 100 users and do 200 rounds per user, where at each round, 300 genre-diverse movies are sampled from the set of available movies. In our experiments, we aim to compare how well our **lpbTS** algorithm performs compared to baselines for the following: i) Settings where movie ratings for users belonging to the same latent state are in the same absolute scale, and when individual ratings could vary in scale for users. ii) How well our algorithm performs with an offline recovered latent preference ordering model, that could have recovery errors. To recover this latent preference ordering model, we use the *"offline estimation"* data, with uniformly collected movie rating logs between 200 and 300 per user, for both settings in i). This also provides insights into the quality of the two-stage latent preference order recovery. Cumulative regret results are reported as averages over user runs, with standard deviation errors. The average movie ratings over users is also reported. To ensure consistency when rating scales vary, the average ratings are standardized with z-score standardization *per user* and errors reported as standard errors of the mean.

C Proof that isotonic regression solves the constrained MLE problem

Proposition 3. *Let $n_a = \sum_{t=1}^{T} \mathbb{1}[a_t = a]$ and define $w_a = \frac{n_a}{\sigma_a^2}$. Next, let $O_z = (o_1, ..., o_k)$ be the preference ordering of latent state z. Then, the solution to the isotonic regression problem with outcomes $y_a = \frac{1}{n_a} \sum_{t:a_t=a} r_t$ and sample weights w_a*

$$\underset{\boldsymbol{\mu} \in \mathbb{R}^d}{\textit{minimize}} \quad \sum_{a=1}^{k} w_a(\mu_a - y_a)^2 \quad \textit{subject to} \quad \mu_{o_k} \leq \mu_{o_{k-1}} \leq ... \leq \mu_{o_1}$$

solves the constrained MLE problem in (3).

Proof. We show that the isotonic regression objective is equal to (3) up to a constant. We have

$$\begin{aligned}\sum_{a=1}^{k} w_a(\mu_a - y_a)^2 &= \sum_{a=1}^{k} w_a(\mu_a^2 - 2y_a\mu_a + y_a^2) = \sum_{a=1}^{k} w_a(\frac{1}{n_a} \sum_{t:a_t=a} [\mu_a^2 - 2r_t\mu_a] + y_a^2) \\ &= \sum_{a=1}^{k} \frac{1}{\sigma^2_a}(\sum_{t:a_t=a} [\mu_a^2 - 2r_t\mu_a + r_t^2] + y_a^2 - \sum_{t:a_t=a} \frac{r_t^2}{n_a}) = \sum_{t=1}^{T} \frac{(\mu_{a_t}^2 - r_t^2)}{\sigma_{a_t}^2} + C \,,\end{aligned}$$

where C is a constant w.r.t. $\boldsymbol{\mu}$. Thus minimizing the LHS and RHS yields the same solution. □

D A note on relative feedback

Dueling bandits (Yue and Joachims, 2009; Sui et al., 2018; Bengs et al., 2021) are the simplest-to-analyze as rewards are observed in the same format as the latent state–as relative preference feedback. Let $\mathcal{D}_T = ((a_t, a'_t, \tilde{r}_t))_{t=1}^{T}$ be a sequence of preference feedback events where $a_t, a'_t \in [m]$ are two competing actions and $\tilde{r}_t \in \{0, 1\}$ indicates which action was preferred. Assuming that there are latent, noisy and continuous rewards r_t, r'_t for the two actions, let $\tilde{R}_t = \mathbb{1}[R_t \geq R'_t]$ indicate noisy preferences for a_t over a'_t and $\tilde{r}_t$ its realization.

If for any latent state $z \in [m]$, there exists an (unknown) margin parameter $\rho > 0$ such that most of the time, with a margin ρ, the reward for a is higher than the reward for a', if a is preferred to a' then,

$$p(\tilde{R}_t = 1 \mid a_t, a'_t, z) \leq \begin{cases} \frac{1}{2} - \rho, & a_t \succeq_z a'_t \\ 1, & \text{otherwise} \end{cases}$$

provides a crude upper bound on the likelihood of a single reward from $\mathcal{D}_T$ under z. In other words, the probability of observing $r_t > r'_t$ is less than $1/2 - \rho$ if the action a'_t has lower rank than a_t. Defining $n_i(z)$ to be the number of observed inversions of the rank imposed by z $n_i(z) = \sum_{t=1}^{T} (\mathbb{1}[I_{a_t}(z) > I_{a'_t}(z)] \neq \mathbb{1}[r_t \leq r'_t])$ then, we can upper bound the full likelihood under z as

$$p(\mathcal{D} \mid z) \leq (\frac{1}{2} - \rho)^{n_i(z)} \cdot 1^{T - n_i(z)} \leq 2^{-n_i(z)} \tag{5}$$

and, likewise, the posterior $p(z \mid \mathcal{D}) \propto p(\mathcal{D} \mid z)p(z)$.

Bounding the posterior as in (5) allows us to rule out candidate latent states as the probability decays with the number of inversions, and a similar posterior sampling algorithm like **lpbTS** for preference feedback can be obtained.

E Absolute to relative feedback

Absolute reward feedback can always be turned into preference feedback. For example, plays and rewards $(a_1, r_1), (a_2, r_2), (a_3, r_3), (a_4, r_4)$ can be paired up consecutively: $(a_1, a_2, \mathbb{1}[r_1 > r_2]), (a_3, a_4, \mathbb{1}[r3 > r4])$. This ensures that different pairs comprise independent events, unlike, say, an all-pairs comparison. However, this procedure is likely very inefficient, statistically. Rewards obtained for each action provide much more information than is used by considering the pairs of most recent actions. Moreover, the algorithm does not rule out playing the same action twice, which means it is prone to getting stuck in local optima.

F Rarity of Similar Preference Orderings in LPB

Observation 1. Rarity of Similar Preference Orderings. Consider k actions in the LPB framework, with O_{z_1} and O_{z_2} as two distinct preference orderings drawn randomly from the set of all permutations of k actions. The probability that O_{z_1} and O_{z_2} differ in exactly two positions is:

$$P(|\{i \mid O_{z_1}(i) \neq O_{z_2}(i)\}| = 2, i \in [k]) = \frac{\binom{k}{2}}{k! - 1}$$

Proof. Consider two distinct random permutations O_{z_1} and O_{z_2} of k actions. We need to find the probability that they differ in exactly two positions, i.e., $|\{i \mid O_{z_1}(i) \neq O_{z_2}(i)\}| = 2, i \in [k]$. Define the relative permutation $\tau = O_{z_2}^{-1} \circ O_{z_1}$. The differing positions are the non-fixed points of τ (where $\tau(i) \neq i$), so τ must be a *transposition*, swapping two elements. The number of possible transpositions is $\binom{k}{2}$. The total number of ordered pairs (O_{z_1}, O_{z_2}) with $O_{z_1} \neq O_{z_2}$ is $k! \cdot (k! - 1)$. For each transposition τ, there are $k!$ pairs where $O_{z_2} = O_{z_1} \circ \tau$, since O_{z_1} can be any permutation. Thus, the number of favorable pairs is $\binom{k}{2} \cdot k!$.

Therefore, the probability is:

$$\frac{\binom{k}{2} \cdot k!}{k! \cdot (k! - 1)} = \frac{\binom{k}{2}}{k! - 1}$$

□

G Expanded Algorithm: lpbTS (Thompson Sampling for Latent Preference Bandits)

We provide an expanded algorithm for **lpbTS** in Algorithm 2.

Algorithm 2 : **lpbTS** (Thompson Sampling for Latent Preference Bandits)

```
Input: Number of arms $k$, latent states $m$, ordering matrix $O \in \mathbb{R}^{m\times k}$, noise $\sigma$
Initialize: For each arm $a \in [k]$, set $N_a = 0$, $S_a = 0$, $\hat{\mu}_a = 0$; for all $z \in [m]$, set $\log P_1(z) = -\log m$; set $t \leftarrow 0$
while True do
    if $t = 0$ then
        Select $A_t \sim \text{Uniform}([k])$
    else
        Compute normalized posterior:
            $$P_t(z) = \frac{\exp(\log P_t(z))}{\sum_{z'=1}^{m} \exp(\log P_t(z'))}.$$
        Sample $B_t \sim P_t(z)$
        Select $A_t = O[B_t, 0]$
    end if
    Observe reward $R_t$
    Update: $N_{A_t} \leftarrow N_{A_t} + 1,\ S_{A_t} \leftarrow S_{A_t} + R_t$
    Set $\hat{\mu}_{A_t} = \frac{S_{A_t}}{N_{A_t}}$ (if $N_{A_t} > 0$)
    for each $z \in [m]$ do                                          ▷ Proposition 2
        Perform isotonic regression on the sequence $\{\hat{\mu}_{O[z,0]}, \dots, \hat{\mu}_{O[z,k-1]}\}$ with weights $\{N_{O[z,0]}, \dots, N_{O[z,k-1]}\}$, obtaining estimates $\hat{\mu}_z[O[z,i]]$ for $i = 0, \dots, k-1$.
    end for
    for all $z \in [m]$ do
        Update:                                                      ▷ (4), (3)
            $$\log P_{t+1}(z) = \log P_t(z) - \frac{\left(R_t - \hat{\mu}_z[A_t]\right)^2}{2\sigma^2}.$$
    end for
    Normalize $\{\log P_{t+1}(z)\}$                                  ▷ *log-sum-exp trick*
    $t \leftarrow t + 1$
end while
```

Computational Complexity. The **lpbTS** algorithm has a computational time complexity of $O(Tmk)$, where T is the time horizon, m is the number of latent states, and k is the number of arms. This complexity is primarily from the $O(mk)$ cost per iteration, due to performing isotonic regression for each of the m states on sequences of length k, repeated over T iterations. The space complexity is $O(mk)$, dominated by the storing of the ordering matrix $O \in \mathbb{R}^{m\times k}$, with additional $O(k + m)$ space for arm and latent state variables being relatively minor.

H Two-Stage Latent Preference Order Recovery

To recover the Latent Preference Order, we rely on methods for extracting preferences from pairwise comparisons (Bradley and Terry, 1952) applied to our setting with latent structure. Vigneau et al. (1999) demonstrated that BTMs can be fit for latent structure with an EM approach, and in Figure 5, we provide an empirical comparison showing that our two-stage recovery compares favorably with EM Latent Preference Order recovery.

The following provides an outline of our Preference Order Recovery

Extracting pairwise comparisons: For N instances, logged data with absolute feedback $\mathcal{D}^{\dagger}{}_{T,i} = \{(A_{t,i}, R_{t,i})\}_{t=1}^{T}, \quad i = 1, \dots, N, A_{t,i} \in [k]$ is used to obtain a dataset incorporating pairwise action comparisons for all instances, captured as $\mathcal{D} = \{(r^{(n)}, Y^{(n)})\}_{n=1}^{N}$, where $r^{(n)} \in \mathbb{R}^k$ are rewards (possibly incomplete), and $Y^{(n)} \in \{0,1\}^{k \times k}$ has $y_{ij}^{(n)} = 1$ if item i beats j in observation n, else 0.

Clustering instances on observed absolute rewards: Observations are then clustered into m groups $\mathcal{D}_1, \dots, \mathcal{D}_m$ using $\{r^{(n)}\}$ (via KMeans with zero imputation for incomplete rewards) so that each n belongs to a cluster $z \in [m]$.

Fitting cluster BTMs: With acces to pairwise comparisons, the preference strengths of the actions in each cluster can be obtained, defined with the *utility*, $\boldsymbol{\beta}^{(z)} = (\beta_0^{(z)}, \dots, \beta_{k-1}^{(z)})$, $\sum_{i=0}^{k-1} \beta_i^{(z)} = 0$ (Bradley and Terry, 1952) where the utility defines the preference strength for each action. We use a Logistic Bradley-Terry model: $P(i \succ j \mid z) = \sigma(\beta_i^{(z)} - \beta_j^{(z)}), \quad \sigma(x) = \frac{1}{1+e^{-x}}$.

Bradley-Terry models (Bradley and Terry, 1952) per cluster are then fit, which give the cluster-data log-likelihood for a cluster z as

$$\ell^{(z)}(\boldsymbol{\beta}^{(z)}) = \sum_{i \neq j} \Big[y_{ij}^{(z)} \ln \sigma(\beta_i^{(z)} - \beta_j^{(z)}) + \big(w_{ij}^{(z)} - y_{ij}^{(z)}\big) \ln \Big(1 - \sigma(\beta_i^{(z)} - \beta_j^{(z)})\Big) \Big].$$

with agregated *cluster* pairwise comparisons

$$y_{ij}^{(z)} = \sum_{n \in \mathcal{D}_z} y_{ij}^{(n)}, \quad w_{ij}^{(z)} = \sum_{n \in \mathcal{D}_z} I_{ij}^{(n)},$$

and where $I_{ij}^{(n)} = 1$ if pair (i, j) is observed in n.

Post-step Sigmoid: The real-valued cluster utilities obtained, $\boldsymbol{\beta}^{(z)}$, are then passed through a Sigmoid to restrict them to $[0,1]$, obtaining $\tilde{\boldsymbol{\beta}}^{(z)} = \sigma(\boldsymbol{\beta}^{(z)}) = \frac{1}{1+e^{-\boldsymbol{\beta}^{(z)}}}$. This Sigmod step also enables us to model cluster *reward mean vectors* $\hat{\mu}^{z}$, by scaling $\tilde{\boldsymbol{\beta}}_i^{(z)}$ to an appropriate range, for example $[1,5]$ for movie ratings with the MovieLens dataset.

A Preference Order O is obtained simply by sorting $\tilde{\boldsymbol{\beta}}^{(z)}$. Algorithm 3 provides a summary for this.

Algorithm 3 Latent Logistic Bradley Terry Model (BTM) for recovering O

```
Require: Data $\{(r^{(n)}, Y^{(n)})\}_{n=1}^{N}$, number of clusters $m$
Cluster $\{r^{(n)}\}_{n=1}^{N}$ into $\mathcal{D}_1, \dots, \mathcal{D}_m$ (via KMeans)        ▷ Pre-step clustering
for $z = 1$ to $m$ do
    for $i, j = 0$ to $k-1$, $i \neq j$ do
        $y_{ij}^{(z)} \leftarrow \sum_{n \in \mathcal{D}_z} y_{ij}^{(n)}$
        $w_{ij}^{(z)} \leftarrow \sum_{n \in \mathcal{D}_z} I_{ij}^{(n)}$
    end for
    $\hat{\boldsymbol{\beta}}^{(z)} \leftarrow \arg\max_{\boldsymbol{\beta}^{(z)}} \ell^{(z)}(\boldsymbol{\beta}^{(z)})$ subject to $\sum_{i=0}^{k-1} \beta_i^{(z)} = 0$
    $\tilde{\boldsymbol{\beta}}^{(z)} \leftarrow \sigma(\hat{\boldsymbol{\beta}}^{(z)})$        ▷ Post-step Sigmoid
    $O_z \leftarrow \mathrm{argsort}(-\tilde{\boldsymbol{\beta}}^{(z)})$
end for
Output: $\{O_z\}_{z=1}^{m}$, $\{\tilde{\boldsymbol{\beta}}^{(z)}\}_{z=1}^{m}$
```

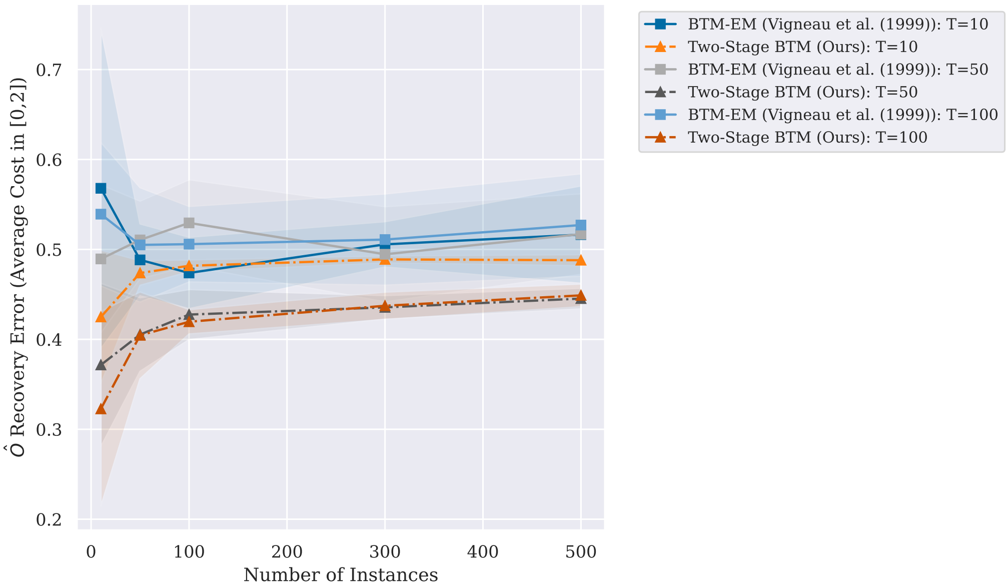
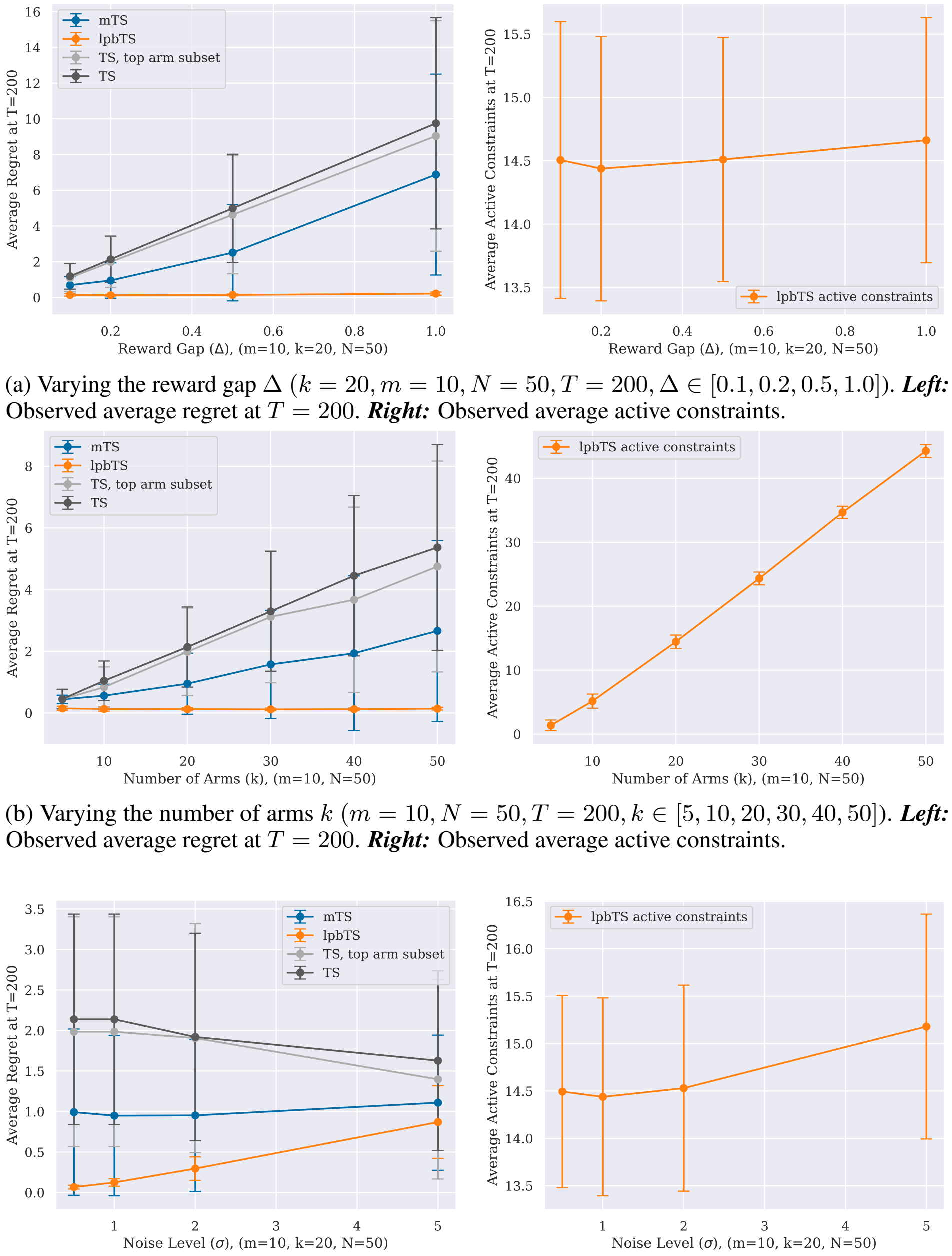


Figure 5: **Synthetic Recovery Experiment (O known, generated according to Section 5)**. Illustration of the average matching error between true and recovered orderings, computed as the average of (1 - Kendall’s tau correlation) after optimal matching using the Hungarian algorithm. The error decreases as the number of instances increases, indicating improved recovery accuracy with more data. Our two-stage recovery compares favorably to an EM approach.

I Additional Experiments

Here we outline additional results from our synthetic ablation study in Figures 6 and 7, and our MovieLens experiments (with 1M Dataset in Figure 8, and 32M Dataset in Figure 9).

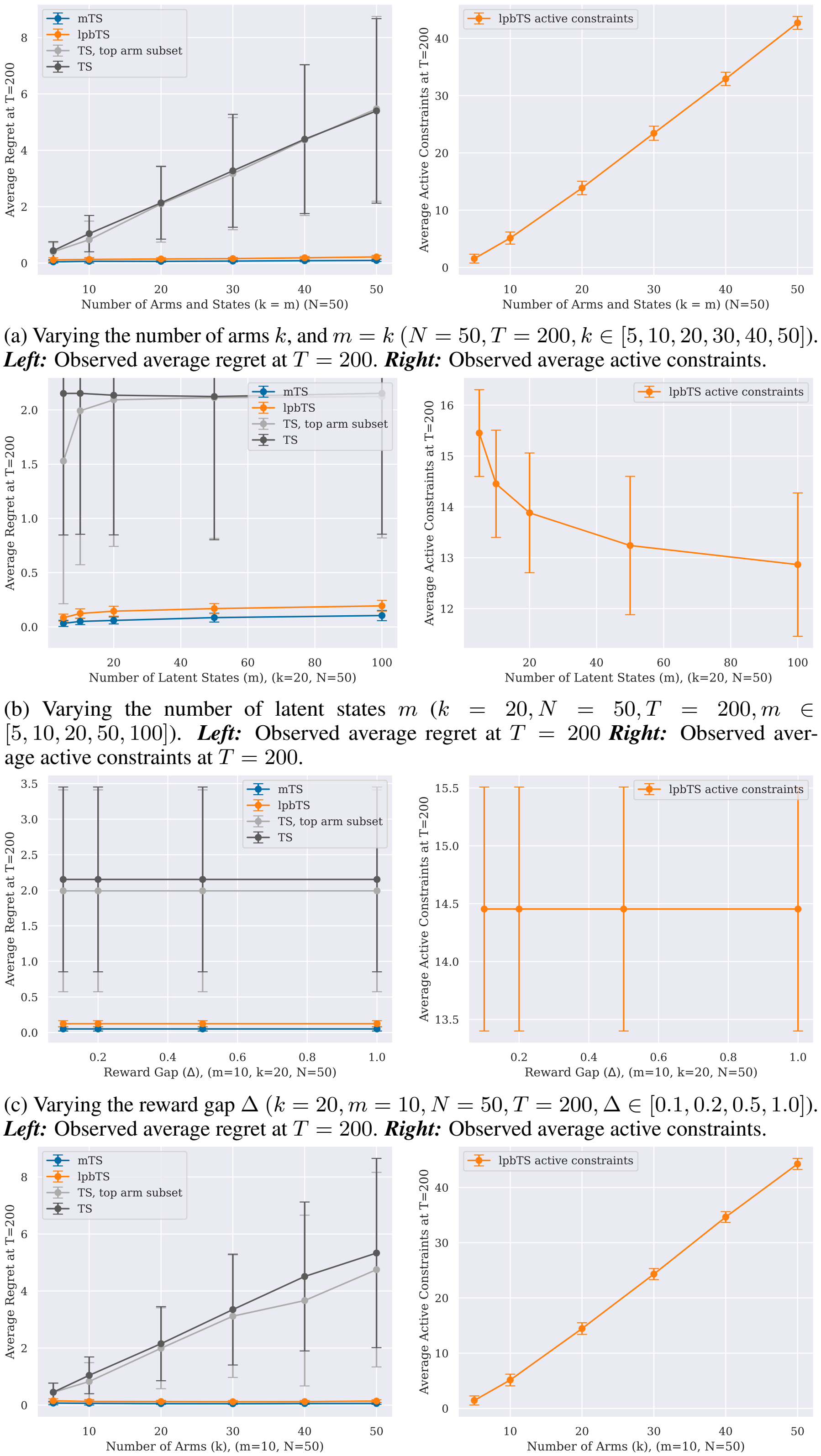


(a) Varying the reward gap Δ ($k = 20, m = 10, N = 50, T = 200, \Delta \in [0.1, 0.2, 0.5, 1.0]$). ***Left:*** Observed average regret at $T = 200$. ***Right:*** Observed average active constraints.

(b) Varying the number of arms k ($m = 10, N = 50, T = 200, k \in [5, 10, 20, 30, 40, 50]$). ***Left:*** Observed average regret at $T = 200$. ***Right:*** Observed average active constraints.

(c) Varying the reward noise σ ($k = 20, m = 10, N = 50, T = 200, \sigma \in [0.5, 1, 2, 5]$). ***Left:*** Observed average regret at $T = 200$. ***Right:*** Observed average active constraints.

Figure 6: Synthetic experiment, instance rewards in different scales. With increasing reward separation Δ, **lpbTS** maintains a stably low regret by leveraging structural constraints and **mTS** regret only worsens because the the draw interval for the mean reward of the optimal arm grows with Δk. The active constraints in **lpbTS** increases as k grows because the number of possible permutations increase like $k!$, so the probability of having large differences between states grows when k grows, and there's need to distinguish more alternative states. When reward noise increases, distinguishing alternative states becomes harder as shown in the increase in active constraints in **lpbTS**.



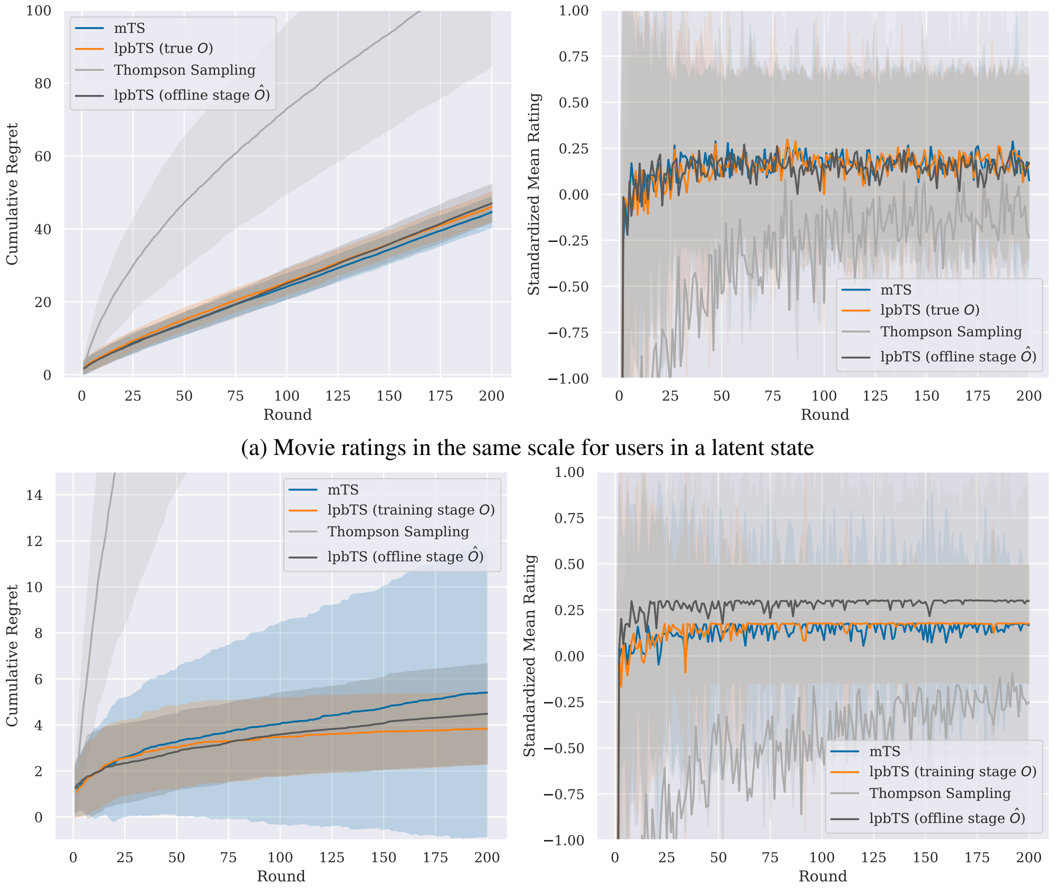
(a) Varying the number of arms k, and $m = k$ ($N = 50, T = 200, k \in [5, 10, 20, 30, 40, 50]$). ***Left:*** Observed average regret at $T = 200$. ***Right:*** Observed average active constraints.

(b) Varying the number of latent states m ($k = 20, N = 50, T = 200, m \in [5, 10, 20, 50, 100]$). ***Left:*** Observed average regret at $T = 200$ ***Right:*** Observed average active constraints at $T = 200$.

(c) Varying the reward gap Δ ($k = 20, m = 10, N = 50, T = 200, \Delta \in [0.1, 0.2, 0.5, 1.0]$). ***Left:*** Observed average regret at $T = 200$. ***Right:*** Observed average active constraints.

(d) Varying the number of arms k ($m = 10, N = 50, T = 200, k \in [5, 10, 20, 30, 40, 50]$). ***Left:*** Observed average regret at $T = 200$. ***Right:*** Observed average active constraints.

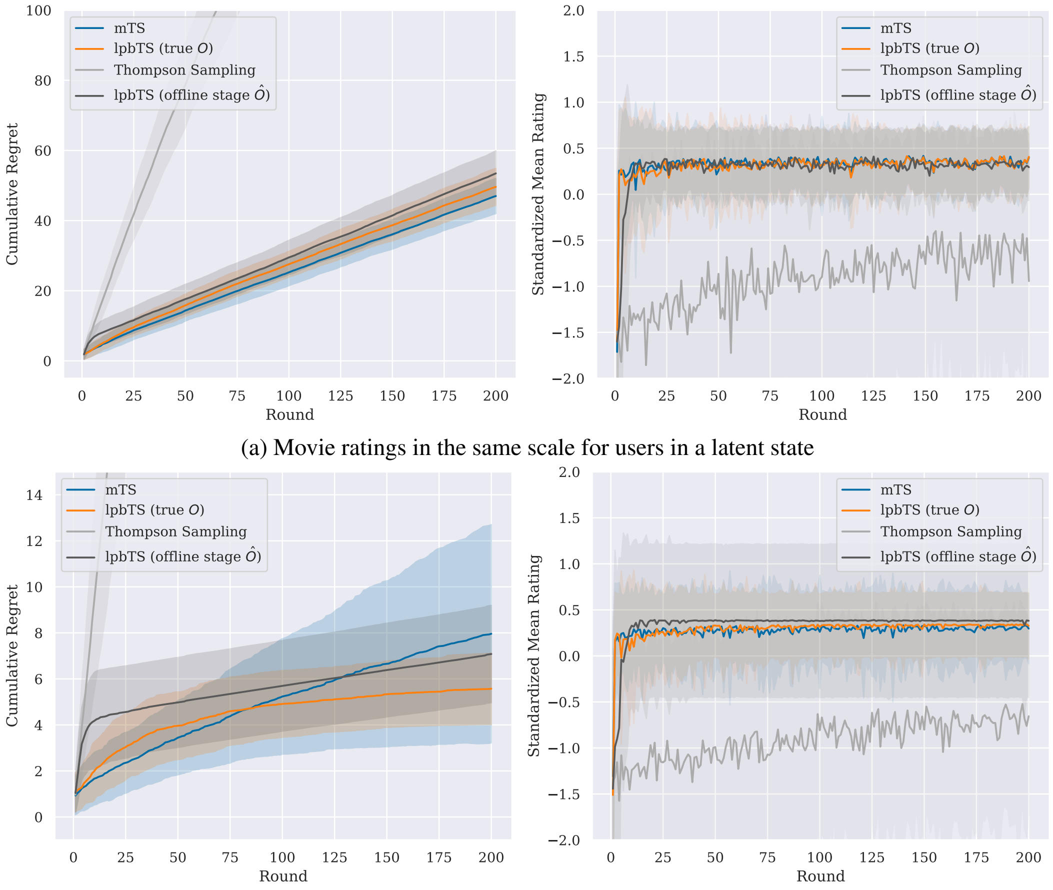
Figure 7: Synthetic experiment, instance rewards in the same scale. Here, **mTS** outperforms in regret due to it's knowledge of true means. Active constraints observed in **lpbTS** explain the latent structure characteristics described earlier more clearly.



(a) Movie ratings in the same scale for users in a latent state

(b) Movie ratings in different scales for users in a latent state

Figure 8: MovieLens Experiment, 1M Dataset. Results match theory: **lpbTS (Ours)** is comparable to mTS in (a), outperforms in (b), and the two-stage recovery of O is empirically validated. Furthermore, we see that when the offline stage $\hat{O}$ is trained with instance reward scales varying, it extracts preferences that are more robust (ground truth O for both was trained with rewards in the same scale).



(a) Movie ratings in the same scale for users in a latent state

(b) Movie ratings in different scales for users in a latent state

Figure 9: MovieLens Experiment, 32M Dataset. Results match theory: **lpbTS (Ours)** is comparable to mTS in (a), outperforms in (b), and the two-stage recovery of O is empirically validated. Furthermore, we see that when the offline stage $\hat{O}$ is trained with instance reward scales varying, it extracts preferences that are more robust (ground truth O for both was trained with rewards in the same scale).

J Computation Infrastructure and Code

The synthetic simulations were run on a 2.6 GHz 6-Core Intel Core i7 Macbook laptop, with 16 GB RAM. Ablation studies and MovieLens experiments were done on a compute cluster with 2 nodes, each having an NVIDIA Tesla T4 GPU with 16GB RAM (Each with 4 Intel Xeon Gold 6226R CPU, 2.90GHz and 72 GB DDR4 RAM).

Code to reproduce the experiments is provided in a separate Zip file.

K Broader Impact

Although the methods in this work are primarily methodological and do not have direct practical implications, they are designed with sequential decision-making in healthcare in mind, and we hope to contribute to speeding up the full impact of machine learning in sequential decision-making in clinical settings. However, any application of our method in general personalization must be made with caution and sufficient guard rails appropriate for the specific problem.