No Regrets for Learning the Prior in Bandits

Anonymous Author(s)
Affiliation
Address
email

Abstract

We propose AdaTS, a Thompson sampling algorithm that adapts sequentially to bandit tasks that it interacts with. The key idea in AdaTS is to adapt to an unknown task prior distribution by maintaining a distribution over its parameters. When solving a bandit task, that uncertainty is marginalized out and properly accounted for. AdaTS is a fully-Bayesian algorithm that can be implemented efficiently in several classes of bandit problems. We derive upper bounds on its Bayes regret that quantify the loss due to not knowing the task prior, and show that it is small. Our theory is supported by experiments, where AdaTS outperforms prior algorithms and works well even in challenging real-world problems.

1 Introduction

We study the problem of maximizing the total reward, or minimizing the total regret, in a sequence of stochastic bandit instances [28, 3, 30]. We consider a Bayesian version of the problem, where the bandit instances are drawn from some distribution. More specifically, the learner interacts with m bandit instances in m tasks, with one instance per task. The interaction with each task is for n rounds and with K arms. The reward distribution of arm $i \in [K]$ in task $s \in [m]$ is $p_i(\cdot; \theta_{s,*})$, where $\theta_{s,*}$ is a shared parameter of all arms in task s. When arm i is pulled in task s, the learner receives a random reward from $p_i(\cdot; \theta_{s,*})$. The parameters $\theta_{1,*}, \ldots, \theta_{m,*}$ are drawn independently of each other from a *task prior* $P(\cdot; \mu_*)$. The task prior is parameterized by an unknown *meta-parameter* μ_*, which is drawn from a *meta-prior* Q. The learner does not know μ_* or $\theta_{1,*}, \ldots, \theta_{m,*}$. However, it knows the parametric forms of all distributions and also knows Q, which allows it to learn about μ_*. This is a form of *meta-learning* [38, 39, 6, 7], where the agent learns to act from interactions with bandit instances.

A simple approach is to ignore the hierarchical structure of the problem and solve each bandit task independently with some bandit algorithm, such as *Thompson sampling (TS)* [37, 10, 2, 36]. This may be highly suboptimal. To illustrate this, imagine that μ_* is such that for any θ in the support of $P(\cdot; \mu_*)$, arm 1 is optimal. If μ_* was known, any reasonable algorithm would only pull arm 1, and thus had the total regret zero over any horizon. Accordingly, we also expect that a clever algorithm that learns μ_* would eventually pull arm 1 an overwhelming fraction of time, and thus suffer regret that diminishes on average as it interacts with a growing number of tasks. Two issues arise when designing the clever algorithm. First, can it be designed computationally efficiently? Second, what is the regret due to adapting to μ_*?

We make the following contributions. First, we propose a Thompson sampling algorithm for our problem, which we call AdaTS. The key idea in our approach is to maintain a distribution over the meta-parameter μ_*, which concentrates over time and is marginalized out when interacting with an individual bandit instance. Second, we propose exact computationally-efficient implementations of AdaTS for multi-armed bandits [28, 3], linear bandits [13, 1], and combinatorial semi-bandits [18, 24, 12]. These implementations are for specific reward distributions and conjugate task priors. Third, we bound the n-round Bayes regret of AdaTS in linear bandits and semi-bandits, and multi-

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

armed bandits as a special case. The Bayes regret is defined by taking expectation over all random quantities, including $\mu_* \sim Q$. Our bounds show that not knowing μ_* has a minimal impact on the regret as the number of tasks grows, of only $\tilde{O}(\sqrt{mn})$. This is in a sharp contrast to prior work [27], where this is $\tilde{O}(n^2\sqrt{m})$. Finally, our experiments show that `AdaTS` quickly adapts to the unknown μ_* and is robust to a misspecified meta-prior Q. We also show that `AdaTS` can adapt in challenging classification problems.

2 Setting

We first introduce our notation. The set $\{1, \dots, n\}$ is denoted by $[n]$. The indicator $\mathbb{1}\{E\}$ denotes that event E occurs. The i-th entry of vector v is v_i, unless the vector or its index are subindexed, in which case we write $v(i)$. We use $\tilde{O}$ for the big-O notation up to polylogarithmic factors. A diagonal matrix with entries v is denoted $\operatorname{diag}(v)$. We use the terms "arm" and "action" interchangeably, based on which better fits a given context.

Our setting was proposed in Kveton et al. [27] and is defined as follows. We have K arms, and each bandit *problem instance* is specified by K distributions $p_1(\cdot;\theta), \dots, p_K(\cdot;\theta)$ over real numbers that share a common parameter $\theta \in \Theta$. The mean of $p_i(\cdot;\theta)$ is denoted by $r(i;\theta)$. The learning agent interacts with m instances, one at each of m *tasks*. At the beginning of task $s \in [m]$, an instance $\theta_{s,*} \in \Theta$ is sampled i.i.d. from a *task prior* $P(\cdot;\mu_*)$, which is parameterized by μ_*. The agent interacts with $\theta_{s,*}$ for n rounds. In round $t \in [n]$, it pulls one arm and observes a stochastic realization of its reward. We denote the pulled arm in round t of task s by $A_{s,t} \in [K]$, the realized rewards of all arms in round t of task s by $Y_{s,t} \in \mathbb{R}^K$, and the reward of arm $i \in [K]$ by $Y_{s,t}(i) \sim p_i(\cdot;\theta_{s,*})$. We assume that the realized rewards $Y_{s,t}$ are i.i.d. with respect to both s and t. A graphical model of our environment is drawn in Figure 1. We define distribution-specific parameters μ_q, Σ_q, Σ_0, and σ^2 when we instantiate our framework. Our terminology is summarized in Appendix A.

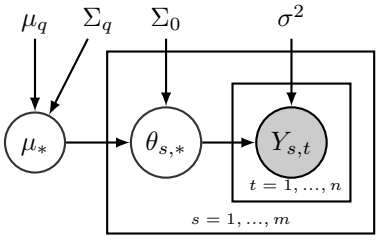

Figure 1: Graphical model of our environment.

The *n-round regret* of an agent or algorithm over m tasks with task prior $P(\cdot;\mu_*)$ is defined as

$$R(m, n; \mu_*) = \sum_{s=1}^{m} \mathbb{E}\left[\sum_{t=1}^{n} r(A_{s,*};\theta_{s,*}) - r(A_{s,t};\theta_{s,*}) \,\middle|\, \mu_*\right], \tag{1}$$

where $A_{s,*} = \arg\max_{i\in[K]} r(i;\theta_{s,*})$ is the *optimal arm* in the random problem instance $\theta_{s,*}$ in task $s \in [m]$. The above expectation is over problem instances $\theta_{s,*} \sim P(\cdot;\mu_*)$, their realized rewards, and also pulled arms. Note that μ_* is fixed. Russo and Van Roy [35] showed that the Bayes regret, which matches the definition in (1) in any task, of Thompson sampling in a K-armed bandit with n rounds is $\tilde{O}(\sqrt{Kn})$. So, when TS is applied independently in each task, $R(m, n; \mu_*) = \tilde{O}(m\sqrt{Kn})$.

Our goal is to attain a comparable regret without knowing μ_*. We frame this problem in a Bayesian fashion, where $\mu_* \sim Q$ before the learning agent interacts with the first task. The agent knows Q and we call it a *meta-prior*. Accordingly, we consider $R(m, n) = \mathbb{E}[R(m, n; \mu_*)]$ as a metric and call it the *Bayes regret*. Our approach is motivated by hierarchical Bayesian models [19], where the uncertainty of prior parameters, such as μ_*, is represented by another distribution, such as Q. In these models, Q is called a *hyper-prior* and μ_* is called *hyper-parameters*. We attempt to learn μ_* from sequential interactions with instances $\theta_{s,*} \sim P(\cdot;\mu_*)$, which are also unknown. The agent can only observe their noisy realizations $Y_{s,t}$.

3 Algorithm

Our algorithm is presented in this section. To describe it, we need to introduce several notions of *history*, the past interactions of the agent. We denote by $H_s = (A_{s,t}, Y_{s,t}(A_{s,t}))_{t=1}^{n}$ the history in task s and by $H_{1:s} = H_1 \oplus \cdots \oplus H_s$ a concatenated vector of all histories in the first s tasks. The history up to round t in task s is $H_{s,t} = (A_{s,\ell}, Y_{s,\ell}(A_{s,\ell}))_{\ell=1}^{t-1}$ and all history up to round t in task s

Algorithm 1 `AdaTS`: Instance-adaptive exploration in Thompson sampling.

Initialize meta-prior $Q_0 \leftarrow Q$
for $s = 1, \dots, m$ **do**
  Compute meta-posterior Q_s (Proposition 1)
  Compute uncertainty-adjusted task prior P_s (Proposition 1)
  for $t = 1, \dots, n$ **do**
    Compute posterior density of θ in task s, $P_{s,t}(\theta) \propto \mathcal{L}_{s,t}(\theta) P_s(\theta)$
    Sample $\tilde{\theta}_{s,t} \sim P_{s,t}$, pull arm $A_{s,t} \leftarrow \arg\max_{i \in [K]} r(i; \tilde{\theta}_{s,t})$, and observe $Y_{s,t}(A_{s,t})$

is $H_{1:s,t} = H_{1:s} \oplus H_{s,t}$. We denote the conditional probability distribution given history $H_{1:s,t}$ by $\mathbb{P}_{s,t}(\cdot) = \mathbb{P}(\cdot \mid H_{1:s,t})$ and the corresponding conditional expectation by $\mathbb{E}_{s,t}[\cdot] = \mathbb{E}[\cdot \mid H_{1:s,t}]$.

The easiest way of explaining our approach is by introducing Thompson sampling [37, 10, 2, 36]. The key idea in TS is to pull arms proportionally to being optimal with respect to the posterior. In particular, fix task s and let $\mathcal{L}_{s,t}(\theta) = \prod_{\ell=1}^{t-1} p_{A_{s,\ell}}(Y_{s,\ell}(A_{s,\ell}); \theta)$ be the *likelihood of observations* up to its round t. Under the assumption that the prior $P(\cdot; \mu_*)$ is known, the posterior of instance θ in round t is $P^{\text{TS}}_{s,t}(\theta) \propto \mathcal{L}_{s,t}(\theta)\, P(\theta; \mu_*)$. TS samples $\tilde{\theta}_t \sim P^{\text{TS}}_{s,t}$ and pulls arm $A_t = \arg\max_{i \in [K]} r(i; \tilde{\theta}_t)$.

We address the case of unknown μ_*. The key idea in our method is to maintain a posterior density of μ_*, which we call a *meta-posterior*. This density represents uncertainty in μ_* given history. In task s, we denote it by Q_s and define it such that $\mathbb{P}(\mu_* \in B \mid H_{1:s-1}) = \int_{\mu \in A} Q_s(\mu)\, d\kappa_1(\mu)$ holds for any set B, where κ_1 is the reference measure for μ. We use this more general notation, as opposing to $d\mu$, because μ can be both continuous and discrete. When solving task s, Q_s used to compute an *uncertainty-adjusted task prior* P_s, which is a posterior density of $\theta_{s,*}$ given history. Formally, P_s is a density such that $\mathbb{P}(\theta_{s,*} \in B \mid H_{1:s-1}) = \int_{\theta \in B} P_s(\theta)\, d\kappa_2(\theta)$ holds for any set B, where κ_2 is the reference measure for θ. After computing P_s, we use TS with a task prior P_s to solve task s. To maintain Q_s and P_s, we find it useful expressing them using a recursive update rule below.

Proposition 1. *Let $\mathcal{L}_s(\theta) = \prod_{\ell=1}^{n} p_{A_{s,\ell}}(Y_{s,\ell}(A_{s,\ell}); \theta)$ be the likelihood of observations in task s. Then for any task $s \in [m]$,*

$$P_s(\theta) = \int_{\mu} P(\theta; \mu)\, Q_s(\mu)\, d\kappa_1(\mu)\,, \quad Q_s(\mu) = \int_{\theta} \mathcal{L}_{s-1}(\theta)\, P(\theta; \mu)\, d\kappa_2(\theta)\, Q_{s-1}(\mu)\,.$$

The above proposition is proved in Appendix A. The pseudocode of our proposed algorithm is in Algorithm 1. We call it `AdaTS` because it *adapts* to the unknown task prior $P(\cdot; \mu_*)$. `AdaTS` can be implemented efficiently when P_s is a conjugate prior for rewards, or a mixture of conjugate priors. We discuss several exact and efficient implementations starting from Section 3.1.

The design of `AdaTS` is motivated by `MetaTS` [27], which also relies on a meta-posterior Q_s. The key difference is that `MetaTS` samples $\tilde{\mu}_s \sim Q_s$ in task s to be optimistic with respect to the unknown μ_*. Then it runs TS with prior $P(\cdot; \tilde{\mu}_s)$. While simple and intuitive, the sampling of $\tilde{\mu}_s$ induces a high variance and leads to a conservative worst-case analysis. We improve `MetaTS` by avoiding the sampling step. This leads to tighter and more general regret bounds (Section 4), beyond multi-armed bandits; while the actual practical performance also improves (Section 5).

3.1 Gaussian Bandit

We start with a K-armed Gaussian bandit with mean arm rewards $\theta \in \mathbb{R}^K$. The reward distribution of arm i is $p_i(\cdot; \theta) = \mathcal{N}(\cdot; \theta_i, \sigma^2)$, where $\sigma > 0$ is reward noise and θ_i is the mean reward of arm i. A natural conjugate prior for this problem class is $P(\cdot; \mu) = \mathcal{N}(\cdot; \mu, \Sigma_0)$, where $\Sigma_0 = \operatorname{diag}\left((\sigma_{0,i})_{i=1}^K\right)$ is known and we learn $\mu \in \mathbb{R}^K$.

`AdaTS` can be implemented efficiently with a Gaussian meta-prior $Q(\cdot) = \mathcal{N}(\cdot; \mu_q, \Sigma_q)$, where both $\mu_q = (\mu_{q,i})_{i=1}^K$ and $\Sigma_q = \operatorname{diag}\left((\sigma_{q,i})_{i=1}^K\right)$ are known. In this case, the meta-posterior in task s is $Q_s(\cdot) = \mathcal{N}(\cdot; \hat{\mu}_s, \hat{\Sigma}_s)$, where $\hat{\mu}_s = (\hat{\mu}_{s,i})_{i=1}^K$ and $\hat{\Sigma}_s = \operatorname{diag}\left((\hat{\sigma}_{s,i})_{i=1}^K\right)$ are defined as

$$\hat{\mu}_{s,i} = \hat{\sigma}^2_{s,i} \left(\frac{\mu_{q,i}}{\sigma^2_{q,i}} + \sum_{\ell=1}^{s-1} \frac{T_{i,\ell}}{T_{i,\ell}\, \sigma^2_{0,i} + \sigma^2} \frac{B_{i,\ell}}{T_{i,\ell}} \right)\,, \quad \hat{\sigma}^{-2}_{s,i} = \sigma^{-2}_{q,i} + \sum_{\ell=1}^{s-1} \frac{T_{i,\ell}}{T_{i,\ell}\, \sigma^2_{0,i} + \sigma^2}\,. \tag{2}$$

Here $T_{i,\ell} = \sum_{t=1}^{n} \mathbb{1}\{A_{\ell,t} = i\}$ is the number of pulls of arm i in task ℓ and the total reward from these pulls is $B_{i,\ell} = \sum_{t=1}^{n} \mathbb{1}\{A_{\ell,t} = i\} Y_{\ell,t}(i)$. The above formula has a very nice interpretation. The posterior mean $\hat{\mu}_{s,i}$ of the meta-parameter of arm i is a weighted sum of the noisy estimates of the means of arm i from the past tasks $B_{i,\ell}/T_{i,\ell}$ and the prior. In this sum, each bandit task is essentially a single observation. The weights are proportional to the number of pulls in a task, giving the task with more pulls a higher weight. They vary from $(\sigma_{0,i}^2 + \sigma^2)^{-1}$, when the arm is pulled only once, up to $\sigma_{0,i}^{-2}$. This is the minimum amount of uncertainty that cannot be reduced by more pulls.

The closed-form update in (2) follows from Lemma 7 (Appendix A) and is due to Kveton et al. [27]. From Proposition 1, we have that the task prior for task s is $P_s(\cdot) = \mathcal{N}(\cdot; \hat{\mu}_s, \hat{\Sigma}_s + \Sigma_0)$.

3.2 Linear Bandit with Gaussian Rewards

Now we generalize Section 3.1 and consider a *linear bandit* [13, 1] with K arms and d dimensions. Let $\mathcal{A} \subset \mathbb{R}^d$ be an *action set* such that $|\mathcal{A}| = K$. We refer to each $a \in \mathcal{A}$ as an *arm*. Then, with a slight abuse of notation from Section 2, the reward distribution of arm a is $p_a(\cdot; \theta) = \mathcal{N}(\cdot; a^\top \theta, \sigma^2)$, where $\theta \in \mathbb{R}^d$ is shared by all arms and $\sigma > 0$ is reward noise. A conjugate prior for this problem class is $P(\cdot; \mu) = \mathcal{N}(\cdot; \mu, \Sigma_0)$, where $\Sigma_0 \in \mathbb{R}^{d \times d}$ is known and we learn $\mu \in \mathbb{R}^d$.

Similarly to Section 3.1, `AdaTS` can be implemented efficiently with a Gaussian meta-prior. The meta-prior is $Q(\cdot) = \mathcal{N}(\cdot; \mu_q, \Sigma_q)$, where $\mu_q \in \mathbb{R}^d$ is a known mean parameter vector and $\Sigma_q \in \mathbb{R}^{d \times d}$ is a known covariance matrix. In this case, $Q_s(\cdot) = \mathcal{N}(\cdot; \hat{\mu}_s, \hat{\Sigma}_s)$, where

$$\hat{\mu}_s = \hat{\Sigma}_s \left(\Sigma_q^{-1} \mu_q + \sum_{\ell=1}^{s-1} \frac{B_\ell}{\sigma^2} - \frac{G_\ell}{\sigma^2} \left(\Sigma_0^{-1} + \frac{G_\ell}{\sigma^2} \right)^{-1} \frac{B_\ell}{\sigma^2} \right),$$

$$\hat{\Sigma}_s^{-1} = \Sigma_q^{-1} + \sum_{\ell=1}^{s-1} \frac{G_\ell}{\sigma^2} - \frac{G_\ell}{\sigma^2} \left(\Sigma_0^{-1} + \frac{G_\ell}{\sigma^2} \right)^{-1} \frac{G_\ell}{\sigma^2}.$$

Here $G_\ell = \sum_{t=1}^{n} A_{\ell,t} A_{\ell,t}^\top$ is the outer product of the feature vectors of the pulled arms in task ℓ and $B_\ell = \sum_{t=1}^{n} A_{\ell,t} Y_{\ell,t}(A_{\ell,t})$ is their sum weighted by their rewards. The above update follows from Lemma 7 (Appendix A) and is due to Kveton et al. [27]. From Proposition 1, we have that the task prior for task s is $P_s(\cdot) = \mathcal{N}(\cdot; \hat{\mu}_s, \hat{\Sigma}_s + \Sigma_0)$.

We note in passing that when $K = d$ and $\mathcal{A}$ is the standard Euclidean basis of $\mathbb{R}^d$, the linear bandit reduces to a K-armed bandit. Since the covariance matrices are unrestricted here, the formulation in this section also shows how to generalize Section 3.1 to arbitrary covariance matrices.

3.3 Semi-Bandit with Gaussian Rewards

A *stochastic combinatorial semi-bandit* [18, 11, 25], or *semi-bandit* for short, is a K-armed bandit where at most $L \leq K$ arms can be pulled in each round. After the arms are pulled, the agent observes their individual rewards and its total reward is the sum of the individual rewards. Semi-bandits can be used to solve online combinatorial problems, such as learning to route.

We consider a Gaussian reward distribution for each arm, as in Section 3.1. The difference in the semi-bandit formulation is that the *action set* is $\mathcal{A} \subseteq \Pi_L(K)$, where $\Pi_L(K)$ is the set of all subsets of $[K]$ of size at most L. In round t of task s, the agents pulls arms $A_{s,t} \in \mathcal{A}$. The meta-posterior is updated analogously to Section 3.1. The difference is that $\mathbb{1}\{A_{\ell,t} = i\}$ becomes $\mathbb{1}\{i \in A_{\ell,t}\}$.

3.4 Exponential-Family Bandit with Mixture Priors

We consider a general K-armed bandit with mean arm rewards $\theta \in \mathbb{R}^K$. The reward distribution of arm i is any one-dimensional exponential-family distribution parameterized by θ_i. In a Bernoulli bandit, this would be $p_i(\cdot; \theta) = \mathrm{Ber}(\cdot; \theta_i)$. A natural prior for this reward model would be a product of per-arm conjugate priors, such as the product of betas for Bernoulli rewards.

An issue with applying `AdaTS` beyond Gaussian priors is that other exponentially-family priors, such as the product of betas, do not have a computationally tractable prior. This issue can be circumvented by restricting `AdaTS` to a finite hypothesis space. Let $\{P(\cdot; j)\}_{j=1}^{L}$ be a set of L potential conjugate

priors, where each $P(\cdot; j)$ is a product of one-dimensional exponential-family priors. A matching meta-prior is $Q(\cdot) = \mathrm{Cat}(\cdot; w_q)$ for some weights $w_q \in \Delta_{L-1}$, where Δ_{L-1} is the L-dimensional simplex and w_q is a vector of initial beliefs into each potential prior.

In this case, $Q_s(j) = \int_\theta \mathcal{L}_{s-1}(\theta) P(\theta; j)\, \mathrm{d}\theta\, Q_{s-1}(j)$ in Proposition 1 can be computed efficiently for any j, because it is a standard conjugate posterior update for a distribution over θ followed by integrating out θ. Moreover, $P_s(\theta) = \sum_{j=1}^{L} Q_s(j) P(\theta; j)$ would be a mixture of exponential-family priors over θ. This is an instance of latent bandits [21]. For this problem class, Thompson sampling can be implemented exactly and efficiently. We do not analyze this setting because prior-dependent Bayes regret bounds for this problem class do not exist yet.

4 Regret Bounds

We first introduce common notation used in our proofs. The action set $\mathcal{A} \subseteq \mathbb{R}^d$ is fixed. Recall that a matrix $X \in \mathbb{R}^{d \times d}$ is *positive semi-definite (PSD)* if it is symmetric and its smallest eigenvalue is non-negative. For a PSD matrix $X \in \mathbb{R}^{d \times d}$, we define $\sigma^2_{\max}(X) = \max_{a \in \mathcal{A}} a^\top X a$. Although $\sigma^2_{\max}(X)$ depends on $\mathcal{A}$, we suppress this dependence because $\mathcal{A}$ is fixed. We denote by $\lambda_1(X)$ the maximum eigenvalue of X and by $\lambda_d(X)$ the minimum eigenvalue of X.

We also need some basic quantities from information theory. For two probability measures, P and Q over a common measurable space, we use $D(P||Q)$ to denote the relative entropy of P with respect to Q, which is infinite when P is not absolutely continuous with respect to Q, and otherwise it is defined as $D(P||Q) = \int \log(\frac{dP}{dQ}) dP$, where dP/dQ is the Radon-Nikodym derivative of P with respect to Q. By slightly abusing notation, we let $P(X)$ denote the probability distribution of a random variable X, $P(X \in \cdot)$. For jointly distributed random variables X and Y, we let $P(X|Y)$ denote the conditional distribution of X given Y, $P(X \in \cdot|Y)$, which is Y-measurable and depends on random Y. The mutual information between X, Y is defined as $I(X;Y) = D(P(X,Y)||P(X)P(Y))$, where $P(X)P(Y)$ denotes the distribution that is obtained as the product of $P(X)$ and $P(Y)$. Intuitively, $I(X;Y)$ measures the amount of information that one can derive from either X or Y about the "other" variable. For jointly distributed X, Y, and Z, we also need the conditional mutual information between X and Y conditioned on Z. We define it as $I(X;Y|Z) = \mathbb{E}[\hat{I}(X;Y|Z)]$, where $\hat{I}(X;Y|Z) = D(P(X,Y|Z)||P(X|Z)P(Y|Z))$ is the "random" conditional mutual information between X and Y given Z. Finally, recall that the chain rule for the random conditional mutual information states that $\hat{I}(X;Y_1,Y_2|Z) = \mathbb{E}[\hat{I}(X;Y_1|Y_2,Z)|Z] + I(X;Y_2|Z)$. When Z is a "trivial" random variable, we get the usual chain rule $I(X;Y_1,Y_2) = I(X;Y_1|Y_2) + I(X;Y_2)$.

4.1 Generic Regret Bound

We start with a generic adaptation of the analysis of Lu and Van Roy [32] to our setting. In round t of task s, we denote the pulled arm by $A_{s,t}$, the observed reward by $Y_{s,t} \sim p_{A_{s,t}}(\cdot; \theta_{s,*})$, and the optimality gap by $\Delta_{s,t} = r(A_{s,*}, \theta_{s,*}) - r(A_{s,t}, \theta_{s,*})$.

For random variables X and Y, let $I_{s,t}(X;Y) = \hat{I}(X;Y|H_{1:s,t})$ denote the random mutual information between X and Y given history $H_{1:s,t}$, which includes all observations from the first $s-1$ tasks and those from the first $t-1$ rounds of task s. Let $\Gamma_{s,t}$ and $\epsilon_{s,t}$ be potentially history-dependent non-negative random variables such that

$$\mathbb{E}_{s,t}[\Delta_{s,t}] \leq \Gamma_{s,t} \sqrt{I_{s,t}(\theta_{s,*}; A_{s,t}, Y_{s,t})} + \epsilon_{s,t} \tag{3}$$

holds almost surely. We want to keep both $\Gamma_{s,t}$ and $\epsilon_{s,t}$ "small". Then the following lemma provides a bound on the total regret over mn rounds of interaction with m tasks.

Lemma 2. *Suppose that (3) holds for all $s \in [m]$ and $t \in [n]$, for some suitably chosen $\Gamma_{s,t}, \epsilon_{s,t} \geq 0$. In addition, let $(\Gamma_s)_{s \in [m]}$ and Γ be non-negative constants such that $\Gamma_{s,t} \leq \Gamma_s \leq \Gamma$ holds for all $s \in [m]$ and $t \in [n]$ almost surely. Then*

$$\mathbb{E}\left[R(n,m)\right] \leq \Gamma \sqrt{mnI(\mu_*; H_{1:m})} + \sum_{s=1}^{m} \Gamma_s \sqrt{nI(\theta_{s,*}; H_s \mid \mu_*, H_{1:s-1})} + \sum_{s=1}^{m} \sum_{t=1}^{n} \mathbb{E}\left[\epsilon_{s,t}\right] .$$

Here the first term expresses the price of not knowing μ_*, while the second term is the regret that one would suffer even if μ_* was known. According to this result, as promised, the price of not knowing μ_* is negligible provided that the mutual information terms grow slowly with n and m. Specifically, we show shortly in linear bandits that $\Gamma_{s,t}$ and $\epsilon_{s,t}$ can be chosen so that the last term of the bound is comparable to the rest, while $\Gamma_{s,t}$ grows slowly with mn. Furthermore, both $I(\theta_{s,*}; H_s \mid \mu_*, H_{1:s-1})$ and $I(\mu_*; H_{1:m})$ are only logarithmic in m and n. The result is a bound where the regret of learning μ_* is $\tilde{O}(\sqrt{mn})$ and that of learning $\theta_{s,*}$ per task is $\tilde{O}(\sqrt{n})$. Now we are ready to prove Lemma 2.

Proof. First, we use the chain rule of random conditional mutual information, and we derive

$$I_{s,t}(\theta_{s,*}; A_{s,t}, Y_{s,t}) \leq I_{s,t}(\theta_{s,*}, \mu_*; A_{s,t}, Y_{s,t}) = I_{s,t}(\mu_*; A_{s,t}, Y_{s,t}) + I_{s,t}(\theta_{s,*}; A_{s,t}, Y_{s,t} \mid \mu_*)\,.$$

Now we take the square root of both sides, apply $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ to the right-hand side, and multiply both sides by $\Gamma_{s,t}$. This yields

$$\Gamma_{s,t}\sqrt{I_{s,t}(\theta_{s,*}; A_{s,t}, Y_{s,t})} \leq \Gamma_{s,t}\sqrt{I_{s,t}(\theta_{s,*}; A_{s,t}, Y_{s,t} \mid \mu_*)} + \Gamma_{s,t}\sqrt{I_{s,t}(\mu_*; A_{s,t}, Y_{s,t})}\,. \tag{4}$$

We start with the first term in (4). Fix task s. From $\Gamma_{s,t} \leq \Gamma_s$, followed by the Cauchy-Schwarz and Jensen's inequalities, we have

$$\mathbb{E}\left[\sum_{t=1}^{n} \Gamma_{s,t}\sqrt{I_{s,t}(\theta_{s,*}; A_{s,t}, Y_{s,t} \mid \mu_*)}\right] \leq \Gamma_s\sqrt{n\mathbb{E}\left[\sum_{t=1}^{n} I_{s,t}(\theta_{s,*}; A_{s,t}, Y_{s,t} \mid \mu_*)\right]}\,.$$

Thanks to $\mathbb{E}\left[I_{s,t}(\theta_{s,*}; A_{s,t}, Y_{s,t} \mid \mu_*)\right] = I(\theta_{s,*}; A_{s,t}, Y_{s,t} \mid \mu_*, H_{1:s,t})$ and the chain rule of mutual information, we have $\mathbb{E}\left[\sum_{t=1}^{n} I_{s,t}(\theta_{s,*}; A_{s,t}, Y_{s,t} \mid \mu_*)\right] = I(\theta_{s,*}; H_s \mid \mu_*, H_{1:s-1})\,.$

Now we consider the last term in (4). We bound $\Gamma_{s,t}$ using Γ, then apply the Cauchy-Schwarz and Jensen's inequalities, and get $\mathbb{E}\left[\sum_{s=1}^{m}\sum_{t=1}^{n} \Gamma_{s,t}\sqrt{I_{s,t}(\mu_*; A_{s,t}, Y_{s,t})}\right] \leq \Gamma\sqrt{mnI(\mu_*; H_{1:m})}\,,$ where we used the chain rule to get

$$I(\mu_*; H_{1:m}) = \sum_{s=1}^{m}\sum_{t=1}^{n} I(\mu_*; A_{s,t}, Y_{s,t} \mid H_{1:s,t}) = \mathbb{E}\left[\sum_{s=1}^{m}\sum_{t=1}^{n} I_{s,t}(\mu_*; A_{s,t}, Y_{s,t})\right]\,.$$

Collecting the terms finishes the proof. □

4.2 Linear Bandit with Gaussian Rewards

We now derive regret bounds for linear bandits (Section 3.2). Without loss of generality, we assume that the actions lie on or inside the unit sphere.

Assumption 1. *The actions have bounded norm* $\max_{a\in\mathcal{A}} \|a\|_2 \leq 1$.

We start by showing that (3) holds for suitably "small" $\Gamma_{s,t}$ and $\epsilon_{s,t}$. In `AdaTS`, in round t of task s, the posterior distribution of $\theta_{s,*}$ is $\mathcal{N}(\hat{\mu}_{s,t}, \hat{\Sigma}_{s,t})$, where

$$\hat{\mu}_{s,t} = \hat{\Sigma}_{s,t}\left((\Sigma_0 + \hat{\Sigma}_s)^{-1}\hat{\mu}_s + \sum_{\ell=1}^{t-1} A_{s,\ell}Y_{s,\ell}\right)\,, \quad \hat{\Sigma}_{s,t}^{-1} = (\Sigma_0 + \hat{\Sigma}_s)^{-1} + \sum_{\ell=1}^{t-1} \frac{A_{s,\ell}A_{s,\ell}^T}{\sigma^2}\,,$$

and $\hat{\mu}_s$ and $\hat{\Sigma}_s$ are defined in Section 3.2. Then, from the properties of Gaussian distributions and that `AdaTS` samples from the posterior, we get a bound on $\Gamma_{s,t}$ and $\epsilon_{s,t}$ as a function of a tunable parameter $\delta \in (0, 1]$.

Lemma 3. *For all tasks $s \in [m]$, rounds $t \in [n]$, and any $\delta \in (0, 1]$, (3) holds almost surely for*

$$\Gamma_{s,t} = 4\sqrt{\frac{\sigma^2_{\max}(\hat{\Sigma}_{s,t})}{\log(1 + \sigma^2_{\max}(\hat{\Sigma}_{s,t})/\sigma^2)}\log\frac{4|\mathcal{A}|}{\delta}}\,, \quad \epsilon_{s,t} = \sqrt{2\delta\sigma^2_{\max}(\hat{\Sigma}_{s,t})}\,.$$

Moreover, for each task s, the following history-independent bound holds almost surely,

$$\sigma^2_{\max}(\hat{\Sigma}_{s,t}) \leq \lambda_1(\Sigma_0)\left(1 + \frac{\lambda_1(\Sigma_q)}{\lambda_1(\Sigma_0) + (s-1)\lambda_1(\Sigma_q)}\right)\,. \tag{5}$$

Lemma 3 is proved in Appendix C.4. By using the bound in (5), we get that $\Gamma_{s,t} = O(\sqrt{\log(1/\delta)})$ and $\epsilon_{s,t} = O(\sqrt{\delta})$. Lemma 3 differs from Lu and Van Roy [32] in two aspects. First, it considers uncertainty in the estimate of μ_* along with $\theta_{s,*}$. Second, it does not require that the rewards are bounded. Our next lemma bounds the mutual information terms in Lemma 2, by exploiting the hierarchical structure of our linear bandit model.

Lemma 4. *For any $H_{1:s,t}$-adapted action-sequence and any $s \in [m]$, we have*

$$I(\theta_{s,*}; H_s \mid \mu_*, H_{1:s-1}) \leq \tfrac{d}{2} \log\left(1 + n\tfrac{\lambda_1(\Sigma_0)}{\sigma^2}\right), \quad I(\mu_*; H_{1:m}) \leq \tfrac{d}{2} \log\left(1 + m\tfrac{\lambda_1(\Sigma_q)}{\lambda_d(\Sigma_0)+\sigma^2/n}\right).$$

Now we are ready to state our final bound, which is proved as follows. We take mutual-information bounds from Lemma 4, and the bounds on $\Gamma_{s,t}$ and $\epsilon_{s,t}$ from Lemma 3, and plug them into our regret decomposition (Lemma 2). Specifically, from Lemma 3, we get for any s and t that $\sigma^2_{\max}(\hat{\Sigma}_{s,t}) \leq (\lambda_1(\Sigma_q) + \lambda_1(\Sigma_0))$, which yields Γ in Lemma 2. On the other hand, Γ_s is directly bounded using the upper bound in (5). The multiplicative $\left(m + \frac{1}{2}\log\left(1 + m\frac{\lambda_1(\Sigma_q)}{\lambda_1(\Sigma_0)}\right)\right)$ below is a result of summing up Γ_s over all tasks.

Theorem 5 (Linear bandit). *The regret of* `AdaTS` *is bounded for any $\delta \in (0, 1]$ as*

$$R(m,n) \leq 4 \underbrace{\sqrt{\frac{\lambda_1(\Sigma_q)+\lambda_1(\Sigma_0)}{\log\left(1+\frac{\lambda_1(\Sigma_q)+\lambda_1(\Sigma_0)}{\sigma^2}\right)}\log(4|\mathcal{A}|/\delta)}\sqrt{mn\tfrac{d}{2}\log\left(1+\frac{m\lambda_1(\Sigma_q)}{\lambda_d(\Sigma_0)+\sigma^2/n}\right)}}_{\textit{Regret for learning } \mu_*} +$$

$$\left(m + \tfrac{1}{2}\log\left(1 + m\tfrac{\lambda_1(\Sigma_q)}{\lambda_1(\Sigma_0)}\right)\right) \times \underbrace{R_\delta(n;\mu_*)}_{\textit{Per-task regret with known } \mu_*}.$$

The above $R_\delta(n; \mu_*)$ is a per-task regret bound for `AdaTS` with known μ_* and is given by

$$R_\delta(n;\mu_*) = 4\sqrt{\frac{\lambda_1(\Sigma_0)}{\log\left(1+\frac{\lambda_1(\Sigma_0)}{\sigma^2}\right)}\log(4|\mathcal{A}|/\delta)}\sqrt{n\tfrac{d}{2}\log\left(1+n\frac{\lambda_1(\Sigma_0)}{\sigma^2}\right)} + n\sqrt{2\delta\lambda_1(\Sigma_0)},$$

where $\delta \in (0, 1]$ is a free parameter. We note that our per-task regret $R_\delta(n; \mu_*)$ has the same prior dependence as in linear bandits with known μ_* [32].

The bound in Theorem 5 is sublinear in n for $\delta = 1/n^2$. It has three terms. The first term can be viewed as the regret due to learning μ_* over all tasks; and is $\tilde{O}(\sqrt{mn})$. The term $mR_\delta(n; \mu_*)$ is the regret for acting in m tasks when μ_* is known; and is $\tilde{O}(m\sqrt{n})$. The last term $(\log m)R_\delta(n; \mu_*)$ is the additional regret for acting in m tasks when μ_* is unknown; and is $\tilde{O}(\sqrt{n})$. Overall, the extra regret due to unknown μ_* is $\tilde{O}(\sqrt{mn})$, and is much lower than the $\tilde{O}(m\sqrt{n})$ regret for solving m tasks with known μ_*. Therefore, we call `AdaTS` is a *no regret algorithm* for meta-learning in linear bandits. The regret for learning μ_* has a prior dependence of $\sqrt{\lambda_1(\Sigma_q) + \lambda_1(\Sigma_0)}$, where $\lambda_1(\Sigma_q)$ is due to the meta-prior and $\lambda_1(\Sigma_0)$ is due to the task prior.

4.3 Semi-Bandit with Gaussian Rewards

In the semi-bandit (Section 3.3), we use the independence of rewards generated by each arm and decompose the per-round regret differently. Let $\Gamma_{s,t}(k)$ and $\epsilon_{s,t}(k)$ be non-negative history-dependent constants, for each arm $k \in [K]$. We denote by (k) the quantities related to arm k. Here the regret decomposes as

$$\mathbb{E}_{s,t}[\Delta_{s,t}] \leq \sum_{k\in[K]} \mathbb{P}_{s,t}(k \in A_{s,t}) \left(\Gamma_{s,t}(k)\sqrt{I_{s,t}(\theta_{s,*}(k); k, Y_{s,t}(k))} + \epsilon_{s,t}(k)\right).$$

The $(k, Y_{s,t}(k))$ term above corresponds to arm k being a part of the action taken in round t of task s. For any $k \in [K]$, due to the chain rule of mutual information we have

$$I_{s,t}(\theta_{s,*}(k); k, Y_{s,t}(k)) \leq I_{s,t}(\theta_{s,*}(k); k, Y_{s,t}(k) \mid \mu_*(k)) + I_{s,t}(\mu_*(k); k, Y_{s,t}(k)).$$

Next we combine the mutual-information terms across all rounds and tasks, similarly to Lemma 2, and bound corresponding $\Gamma_{s,t}(k)$ and $\epsilon_{s,t}(k)$ independently of m and n. This leads to Theorem 6, which is proved in Appendix D.

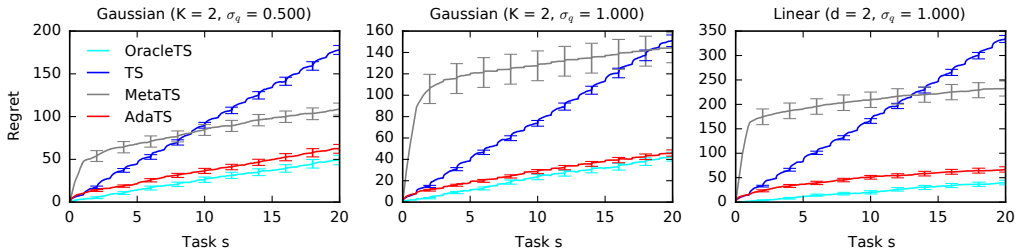

Figure 2: Comparison of AdaTS to three baselines for various meta-prior widths σ_q.

Theorem 6 (**Semi-bandit**). *The regret of* AdaTS *is bounded for any* $\delta \in (0, 1]$ *as*

$$R(m,n) \le 4 \underbrace{\sqrt{\frac{1}{K}\sum_{k\in[K]} \frac{\sigma_{0,k}^2+\sigma_{q,k}^2}{\log\left(1+\frac{\sigma_{0,k}^2+\sigma_{q,k}^2}{\sigma^2}\right)} \log\left(1+m\frac{\sigma_{q,k}^2}{\sigma_{0,k}^2+\sigma^2/n}\right)} \log(\tfrac{4K}{\delta})\sqrt{mnKL}}_{\textit{Regret for learning } \mu_*} +$$

$$\left(m + \tfrac{1}{2}\log\left(1 + m \max_{k\in[K]:\sigma_{0,k}^2>0} \frac{\sigma_{q,k}^2}{\sigma_{0,k}^2}\right)\right) \times \underbrace{R_\delta(n;\mu_*)}_{\textit{Per-task regret with known } \mu_*},$$

where the per-task regret is

$$R_\delta(n;\mu_*) = 4\sqrt{\frac{1}{K}\sum_{k\in[K]} \frac{\sigma_{0,k}^2}{\log\left(1+\frac{\sigma_{0,k}^2}{\sigma^2}\right)} \log\left(1+n\frac{\sigma_{0,k}^2}{\sigma^2}\right)} \log(\tfrac{4K}{\delta})\sqrt{nKL} + n\sqrt{2\delta\frac{1}{K}\sum_{k\in[K]}\sigma_{0,k}^2}\,.$$

The prior widths $\sigma_{q,k}$ *and* $\sigma_{0,k}$ *are defined as in Section 3.1.*

The bound in Theorem 6 is sublinear in n for $\delta = 1/n^2$. Its form resembles Theorem 5. Specifically, the regret for learning μ_* is $\tilde{O}(\sqrt{mnKL})$ and the additional regret for acting under unknown μ_* is $\tilde{O}(\sqrt{n})$. Both of these are much lower than the regret for acting in m tasks when μ_* is known, which is $\tilde{O}(m\sqrt{nKL})$. Thus AdaTS is a *no regret algorithm* for semi-bandits.

Theorem 6 improves upon a naive application of Theorem 5 to semi-bandits. This is because all prior width constants are averages, as opposing to the maximum over arms in Theorem 5. To the best of our knowledge, such per-arm prior-dependence has not been captured in semi-bandits by any prior work. To emphasize the difference, we consider a problem where $\sigma_{0,k}^2 > 0$ for only $K' \ll K$ arms. This means that only K' arms are uncertain in the tasks. In that scenario, the bound in Theorem 6 scales as $\tilde{O}(m\sqrt{nK'L})$, while the bound in Theorem 5 would be $\tilde{O}(m\sqrt{nKL})$.

5 Experiments

We experiment with two synthetic problems. In both problems, the number of tasks is $m = 20$ and each task has $n = 200$ rounds. The first problem is a Gaussian bandit (Section 3.1) with $K = 2$ arms. The meta-prior is $\mathcal{N}(\mathbf{0}, \Sigma_q)$ with $\Sigma_q = \sigma_q^2 I_K$, the prior covariance is $\Sigma_0 = \sigma_0^2 I_K$, and the reward noise is $\sigma = 1$. We experiment with $\sigma_q \ge 0.5$ and $\sigma_0 = 0.1$. Since $\sigma_q \gg \sigma_0$, the entries of $\theta_{s,*}$ are likely to have the same order as those of μ_*. Therefore, a clever agent that learns μ_* could have very low regret. The second problem is a linear bandit (Section 3.2) in $d = 2$ dimensions with $K = 5d$ arms. The action set is sampled from $[-0.5, 0.5]^d$. The meta-prior, prior, and noise are the same as in the Gaussian bandit. All results are averaged over 100 runs.

AdaTS is compared to three baselines. The first is idealized TS with the true prior $\mathcal{N}(\cdot; \mu_*, \Sigma_0)$ and we call it OracleTS. OracleTS shows the minimum attainable regret. The second is agnostic TS, which ignores the structure of the problem. We call it TS and implement it with prior $\mathcal{N}(\mathbf{0}, \Sigma_q + \Sigma_0)$, since $\theta_{s,*}$ can be viewed as a sample from this distribution when the structure is ignored. The third baseline is MetaTS of Kveton et al. [27]. All methods are evaluated by their cumulative regret up to

task s, which we plot as it accumulates round-by-bound within each task (Figure 2). The regret of algorithms that do not learn μ_*, such as `OracleTS` and `TS`, is expected to be linear in s, since they solve s similar tasks with the same policy (Section 2); with a lower slope being a better policy. As no algorithm can outperform `OracleTS`, no regret can grow sublinearly in s.

Our results are reported in Figure 2. We start with a Gaussian bandit with $\sigma_q = 0.5$. This setting is identical to Figure 1b of Kveton et al. [27]. We observe that `AdaTS` outperforms `TS`, which does not learn μ_*, and is comparable to `OracleTS`, which knows μ_*. Its regret is about 30% lower than that of `MetaTS`. Now we increase the meta-prior width to $\sigma_q = 1$. In this setting, meta-posterior sampling in `MetaTS` leads to high biases in earlier tasks. This leads to a major increase in regret; while `AdaTS` performs comparably to `OracleTS`. We end with a linear bandit with $\sigma_q = 1$. In this experiment, `AdaTS` outperforms `MetaTS` again and has more than three times lower regret.

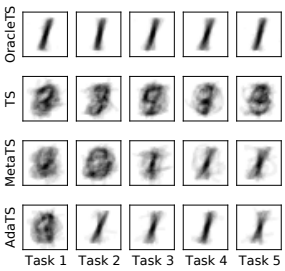

Figure 3: Meta-learning of a highly rewarding digit 1.

Appendix E contains more experiments. In Appendix E.1, we experiment with more values of K and d, and show the robustness of `AdaTS` to missspecified meta-prior Q. In Appendix E.2, we apply `AdaTS` to bandit classification problems. In Figure 3, we show results for one of these problems, meta-learning a highly rewarding digit 1 in the bandit setting. For each method and task s, we show the average digit, corresponding to the pulled arms, in round 1 of task s. `AdaTS` learns a good meta-parameter μ_* almost instantly, since its average digit in task 2 already resembles digit 1.

6 Related Work

Two closest related works are Bastani et al. [5] and Kveton et al. [27]. Bastani et al. [5] proposed Thompson sampling that learns the prior from a sequence of pricing experiments. The algorithm is tailored to pricing and learns through forced exploration using conservative TS. Therefore, it is conservative. They also do not prove prior-dependent bounds. Our setting is identical to Kveton et al. [27]. The novelty stems from our algorithm. While `MetaTS` [27] samples the meta-parameter from its posterior distribution, `AdaTS` marginalizes it out when computing an uncertainty-adjusted task prior. This leads to an approach that is more principled, comparably general, has a fully-Bayesian analysis beyond multi-armed bandits, and may have several-fold lower regret in practice.

`AdaTS` is a meta-learning algorithm [38, 39, 6, 7, 16, 17]. Meta-learning has a long history in multi-armed bandits. Some of the first works are Azar et al. [4] and Gentile et al. [20], who proposed UCB algorithms for multi-task learning. Deshmukh et al. [14] studied multi-task learning in contextual bandits. Cella et al. [9] proposed a UCB algorithm that meta-learns the mean parameter vector in a linear bandit, which is akin to learning μ_* in Section 3.2. Another recent work is Yang et al. [41], who studied regret minimization with multiple parallel bandit instances, with the goal of learning their shared subspace. All of these works are frequentist and analyze a stronger notion of regret. On the other hand, by leveraging the fundamentals of Bayesian reasoning, we end up with practical and general-purpose algorithm designs that explore naturally.

Several recent papers approached the problem of learning a bandit algorithm using policy gradients [15, 8, 26, 42, 34], including learning Thompson sampling [26, 34]. These works focus on offline optimization against a known bandit-instance distribution and have no convergence guarantees in general [8, 26]. Tuning of bandit algorithms is known to reduce regret [40, 33, 23, 22]. Typically it is ad-hoc and we believe that meta-learning is the right way of framing this problem.

7 Conclusions

We propose `AdaTS`, a fully-Bayesian algorithm for meta-learning in bandits that adapts to a sequence of bandit tasks that it interacts with. `AdaTS` attains low regret by adapting the uncertainty in both the meta and per-task parameters. We analyze the Bayes regret of `AdaTS` using information-theory tools that isolate the effect of learning the meta-parameter from that of learning the per-task parameters. For linear bandits and semi-bandits, we derive novel prior-dependent bounds that show that the price of not knowing the meta-parameter is small. Our experiments underscore the wide applicability of `AdaTS`, good out-of-the-box performance, and robustness to meta-prior misspecification.

References

[1] Yasin Abbasi-Yadkori, David Pal, and Csaba Szepesvari. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24*, pages 2312–2320, 2011.

[2] Shipra Agrawal and Navin Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. In *Proceeding of the 25th Annual Conference on Learning Theory*, pages 39.1–39.26, 2012.

[3] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002.

[4] Mohammad Gheshlaghi Azar, Alessandro Lazaric, and Emma Brunskill. Sequential transfer in multi-armed bandit with finite set of models. In *Advances in Neural Information Processing Systems 26*, pages 2220–2228, 2013.

[5] Hamsa Bastani, David Simchi-Levi, and Ruihao Zhu. Meta dynamic pricing: Transfer learning across experiments. *CoRR*, abs/1902.10918, 2019. URL https://arxiv.org/abs/1902.10918.

[6] Jonathan Baxter. Theoretical models of learning to learn. In *Learning to Learn*, pages 71–94. Springer, 1998.

[7] Jonathan Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

[8] Craig Boutilier, Chih-Wei Hsu, Branislav Kveton, Martin Mladenov, Csaba Szepesvari, and Manzil Zaheer. Differentiable meta-learning of bandit policies. In *Advances in Neural Information Processing Systems 33*, 2020.

[9] Leonardo Cella, Alessandro Lazaric, and Massimiliano Pontil. Meta-learning with stochastic linear bandits. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[10] Olivier Chapelle and Lihong Li. An empirical evaluation of Thompson sampling. In *Advances in Neural Information Processing Systems 24*, pages 2249–2257, 2012.

[11] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework, results and applications. In *Proceedings of the 30th International Conference on Machine Learning*, pages 151–159, 2013.

[12] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *CoRR*, abs/1407.8339, 2014. URL http://arxiv.org/abs/1407.8339.

[13] Varsha Dani, Thomas Hayes, and Sham Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 355–366, 2008.

[14] Aniket Anand Deshmukh, Urun Dogan, and Clayton Scott. Multi-task learning for contextual bandits. In *Advances in Neural Information Processing Systems 30*, pages 4848–4856, 2017.

[15] Yan Duan, John Schulman, Xi Chen, Peter Bartlett, Ilya Sutskever, and Pieter Abbeel. RL^2: Fast reinforcement learning via slow reinforcement learning. *CoRR*, abs/1611.02779, 2016. URL http://arxiv.org/abs/1611.02779.

[16] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135, 2017.

[17] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems 31*, pages 9537–9548, 2018.

[18] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking*, 20(5):1466–1478, 2012.

[19] Andrew Gelman, John Carlin, Hal Stern, David Dunson, Aki Vehtari, and Donald Rubin. *Bayesian Data Analysis*. Chapman & Hall, 2013.

[20] Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. In *Proceedings of the 31st International Conference on Machine Learning*, pages 757–765, 2014.

[21] Joey Hong, Branislav Kveton, Manzil Zaheer, Yinlam Chow, Amr Ahmed, and Craig Boutilier. Latent bandits revisited. In *Advances in Neural Information Processing Systems 33*, 2020.

[22] Chih-Wei Hsu, Branislav Kveton, Ofer Meshi, Martin Mladenov, and Csaba Szepesvari. Empirical Bayes regret minimization. *CoRR*, abs/1904.02664, 2019. URL http://arxiv.org/abs/1904.02664.

[23] Volodymyr Kuleshov and Doina Precup. Algorithms for multi-armed bandit problems. *CoRR*, abs/1402.6028, 2014. URL http://arxiv.org/abs/1402.6028.

[24] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Combinatorial cascading bandits. In *Advances in Neural Information Processing Systems 28*, pages 1450–1458, 2015.

[25] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, 2015.

[26] Branislav Kveton, Martin Mladenov, Chih-Wei Hsu, Manzil Zaheer, Csaba Szepesvari, and Craig Boutilier. Differentiable meta-learning in contextual bandits. *CoRR*, abs/2006.05094, 2020. URL http://arxiv.org/abs/2006.05094.

[27] Branislav Kveton, Mikhail Konobeev, Manzil Zaheer, Chih-Wei Hsu, Martin Mladenov, Craig Boutilier, and Csaba Szepesvari. Meta-thompson sampling. *CoRR*, abs/2102.06129, 2021. URL https://arxiv.org/abs/2102.06129.

[28] T. L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[29] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[30] Tor Lattimore and Csaba Szepesvari. *Bandit Algorithms*. Cambridge University Press, 2019.

[31] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[32] Xiuyuan Lu and Benjamin Van Roy. Information-theoretic confidence bounds for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[33] Francis Maes, Louis Wehenkel, and Damien Ernst. Meta-learning of exploration/exploitation strategies: The multi-armed bandit case. In *Proceedings of the 4th International Conference on Agents and Artificial Intelligence*, pages 100–115, 2012.

[34] Seungki Min, Ciamac Moallemi, and Daniel Russo. Policy gradient optimization of Thompson sampling policies. *CoRR*, abs/2006.16507, 2020. URL http://arxiv.org/abs/2006.16507.

[35] Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

[36] Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A tutorial on Thompson sampling. *Foundations and Trends in Machine Learning*, 11(1):1–96, 2018.

[37] 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, 1933.

[38] Sebastian Thrun. *Explanation-Based Neural Network Learning - A Lifelong Learning Approach*. PhD thesis, University of Bonn, 1996.

[39] Sebastian Thrun. Lifelong learning algorithms. In *Learning to Learn*, pages 181–209. Springer, 1998.

[40] Joannes Vermorel and Mehryar Mohri. Multi-armed bandit algorithms and empirical evaluation. In *Proceedings of the 16th European Conference on Machine Learning*, pages 437–448, 2005.

[41] Jiaqi Yang, Wei Hu, Jason Lee, and Simon Du. Provable benefits of representation learning in linear bandits. *CoRR*, abs/2010.06531, 2020. URL http://arxiv.org/abs/2010.06531.

[42] Kaige Yang and Laura Toni. Differentiable linear bandit algorithm. *CoRR*, abs/2006.03000, 2020. URL http://arxiv.org/abs/2006.03000.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Most related works are frequentist and analyze a stronger notion of regret. On the other hand, by leveraging the fundamentals of Bayesian reasoning, we end up with practical and general-purpose algorithm designs that explore naturally.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We develop new bandit algorithms that adapt to a sequence of bandit tasks. Bandit algorithms are often used in user-facing domains. However, we do not propose any new domain. So our impact on users should be limited to those that already exist. Beyond this, we are not aware of any societal consequences of our work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] The setting is described in Section 2 and most theoretical results are in Section 4.
   (b) Did you include complete proofs of all theoretical results? [Yes] The complete proofs are in appendices.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] For synthetic experiments.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] For all experiments.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In all plots.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] The experiments are not computationally heavy.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We use MNIST and Omniglot datasets.
   (b) Did you mention the license of the assets? [No] Our datasets are well known and public.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]