Multi-Environment Meta-Learning in Stochastic Linear Bandits

Ahmadreza Moradipari[1], Mohammad Ghavamzadeh[2], Taha Rajabzadeh[3],
Christos Thrampoulidis[4], Mahnoosh Alizadeh[1]

***Abstract*—In this work we investigate meta-learning (or learning-to-learn) approaches in multi-task linear stochastic bandit problems that can originate from multiple environments. Inspired by the work of [1] on meta-learning in a sequence of linear bandit problems whose parameters are sampled from a single distribution (i.e., a single environment), here we consider the feasibility of meta-learning when task parameters are drawn from a mixture distribution instead. For this problem, we propose a regularized version of the OFUL algorithm that, when trained on tasks with labeled environments, achieves low regret on a new task without requiring knowledge of the environment from which the new task originates. Specifically, our regret bound for the new algorithm captures the effect of environment misclassification and highlights the benefits over learning each task separately or meta-learning without recognition of the distinct mixture components.**

***Index Terms*—Meta-learning, Linear Stochastic Bandit, Sequential Decision Making**

I. Introduction

Stochastic bandit optimization algorithms have long found applications in many fields where some characteristics of the users' response are not known and can only be learnt through a limited number of noisy observations, including recommendation engines, advertisement placement, personalized medicine, etc. The learner's objective for the overall learning task consists of maximizing the cumulative reward gained during T rounds of interaction with the user. The expected reward gained at each round t is a function $f(x_t)$ of the action x_t that the learner chooses to play, and f is not known to the learner. There is a rich literature covering parametric or non-parametric characterizations of f, as well as finite or continuous action sets. An important and widely applicable case is the stochastic linear bandit (LB) problem, where the expected reward is linear in the action x_t, i.e., $f(x_t) = x_t^T\theta$, with θ denoting an unknown vector that describes the users' characteristics.

Now consider the scenario where a recommendation system consecutively deals with users whose characteristics (e.g., the parameter vector θ in the LB case) originate from an *unknown* probability distribution ρ. A classical bandit algorithm would approach each learning task independently, which would translate to high exploration cost to estimate θ for each user. Motivated by this, the work of [1] explores the idea of transferring knowledge between consecutive tasks by designing a meta-learning algorithm for the LB problem. Meta-learning approaches, recently made popular in the reinforcement learning literature in order to address sample complexity issues, allow algorithms to acquire inductive biases in a data-driven manner in order to adapt faster to new situations based on their limited past experience.

In this work we consider the case where the user population's preferences originate from a mixture model, with sub-populations that have distinct (unknown) preference distributions. In this case, we say that the consecutive learning tasks originate from multiple environments. Consider for example a recommendation system where men and women may have distinct preferences (e.g. on Netflix, some movies are seen to be preferred by more women and others by more men). Focused on the LB problem, we show that if the sub-populations' preference distributions are sufficiently distinct, in order to best transfer knowledge across learning tasks, the meta-learning algorithm should first estimate the environment from which the task originates. Our proposed algorithm MEML-OFUL, and the corresponding regret guarantees, formalize the trade-offs associated with this design choice.

Before formally stating the problem, let us provide an overview of prior art under three relevant categories.

Multi-armed Bandits (MAB). Two popular algorithms exist for MAB: 1) the upper confidence bound (UCB) algorithm based on the optimism in the face of uncertainty (OFU) principle [2]–[4], which chooses the best feasible environment and corresponding optimal action at each time step with respect to confidence regions on the unknown parameter; 2) Thompson Sampling (TS) algorithm (a.k.a., posterior sampling), [5]–[7] which samples an environment from the prior at each time step and selects the optimal action with respect to the sampled parameter. For the stochastic Linear bandit (LB) problems, there exist two well-known algorithms for LB are: OFUL or Linear UCB (LinUCB) and Linear Thompson Sampling (LinTS). [8]–[11] provided a regret bound of order $\mathcal{O}(\sqrt{T}\log T)$ for OFUL algorithm and [12]–[15] provided a regret bound of order $\mathcal{O}(\sqrt{T}\log^{3/2} T)$ for LinTS in a frequentist setting.

Multi-task Leaning and Meta-learning. There has been an increasing attention on theoretically studying the ability of a learner to transfer knowledge between different learning tasks, commonly referred to *transfer learning* and applied to both the multi-task learning problem [16]–[21] and the meta-learning problem [22]–[28] in the past years. In particular, the goal of

This work is supported by NSF grant 1847096. C. Thrampoulidis was partially supported by the NSF under Grant Number 1934641.
[1]University of California, Santa Barbara, [2]Google Research, [3]Stanford University, [4]University of British Columbia. Corresponding author: ahmadreza_moradipari@ucsb.edu

multi-task learning is to design an algorithm that performs well on a group of (possibly concurrent) tasks that share a similar representation (e.g., low-dimensional linear representation). The goal of the meta-learning is to select an algorithm that can utilize a number of training tasks from a common environment in order to rapidly adapt to the new task that shares the same environment with the training tasks. We focus on the latter in this paper. Recently, there have been a few works that study the multi-tasks learning in the bandit framework [29]–[36]. In particular, the recent works of [37]–[39] study meta-learning in multi-armed bandits problem with a Bayesian approach. In their settings, they consider a mixture Gaussian distribution as a prior distribution and propose a Thompson-Sampling algorithm with provable regret guarantees. However, in this work, we study a frequentist version of this problem and we propose a UCB-based algorithm. We also consider more general families of distribution for the mixture model.

II. Problem Formulation

In this section, we briefly recall the preliminaries on stochastic linear bandit (LB) problem and previous results on meta-learning in LB, and then we present the multi-environment meta-learning setting considered in this work.

A. The Linear Stochastic Bandit (LB) Problem

In the LB problem, at each round $t \in [T]$, the learner is given an action set $\mathcal{D}_t \subseteq \mathbb{R}^d$ from which she chooses an action $x_t \in \mathcal{D}_t$ and observes a random reward

$$y_t = x_t^\top \theta + \xi_t. \tag{1}$$

In (1), the parameter vector $\theta \in \mathbb{R}^d$ is an *unknown* but fixed reward parameter and ξ_t is zero-mean additive noise. If provided with the knowledge of the true reward parameter θ, the optimal policy at each round t is to play the optimal action $x_t^\star = \arg\max_{x \in \mathcal{D}_t} x^\top \theta$ that maximizes the instantaneous reward. However, in the absence of such knowledge, the goal of the learner is to collect as much reward as possible, or minimise the *cumulative pseudo*-regret up to round T:

$$R(T, \theta) = \sum_{t=1}^{T} x_\star^\top \theta - x_t^\top \theta. \tag{2}$$

This classical setup defines *a single learning task* that takes T rounds to complete. Next, we will explore the setting where the learner is presented with *a sequence of learning tasks* in the form of LB problems that share probabilistic models for the unknown parameter vector θ. By leveraging the structure shared between consecutive tasks, a so-called meta-learning algorithm introduces new inductive biases in the LB problem that allow the learner to transfer knowledge to future tasks.

B. Meta-Learning in LB

The problem of meta-learning for the linear bandit problem was first introduced by [1]. Their meta-learning setting consists of a sequence of consecutive linear bandit problems that share the same *environment*, i.e., their parameter vectors $\theta_1, \ldots, \theta_N, \ldots$ are sampled independently from a task-distribution ρ with a bounded support in $\mathbb{R}^d$ that is unknown to the learner. The learner's goal is to leverage the task similarities (i.e., the fact that they share the same environment) in order to minimize the regret for a new task. In particular, [1] designed an algorithm that achieves a low regret on any new task after being trained over the data provided by N completed tasks. The goal is to control the so-called transfer regret incurred on the $(N+1)$-st task, defined as:

$$\mathcal{R}(T) = \mathbb{E}_{\theta \sim \rho}\left[\mathbb{E}\left[R(T, \theta)\right]\right]. \tag{3}$$

Next we present the setting that we consider in this work, which is an extension of this setting to include multiple environments.

C. Multi-Environment Meta-Learning in LB

In the multi-environment setting, we assume that the consecutive tasks $i = 1, \ldots, N$ originate from one of m environments $\nu = 1, 2, \ldots, m$ following a known multinomial distribution with probabilities $(p_1, p_2, \ldots, p_m)$. Conditioned on the environment ν, the task distribution for the parameter vector θ is denoted as ρ_ν. The distributions ρ_ν have bounded supports in $\mathbb{R}^d$ and are not known to the learner. Instead of approaching each learning task independently, the learner collects information while interacting with the environments over N consecutive tasks in order to perform meta-learning. Specifically, after completing the i-th task, we store the whole interaction in a dataset $\mathcal{Z}_i = \{(x_{i,t}, y_{i,t})\}_{t=1}^{T}$. Then, using the collected datasets from the first N completed tasks, our goal is to design an algorithm that minimizes the regret for a new task with parameter θ_{N+1}, without knowing the environment ν from which θ_{N+1} is sampled. In other words, we wish to design an algorithm that, after being trained over N datasets, leverages its past observations in order to introduce inductive biases to minimize the so-called *transfer-regret* for task $N+1$:

$$\begin{aligned}\mathcal{R}(T) &= \mathbb{E}_{\nu}\left[\mathbb{E}_{\theta \sim \rho_\nu}\left[\mathbb{E}\left[R(T, \theta)\right]\right]\right] &\quad (4)\\ &= \sum_{i=1}^{m} \mathbb{E}_{\theta \sim \rho_\nu}\left[\mathbb{E}\left[R(T, \theta)\right] | \nu = i\right] p_i. &\quad (5)\end{aligned}$$

In (4), the outer expectation is with respect to the randomness over set of possible environments, the middle expectation is with respect to the the task parameters in each environment, and the inner expectation is with respect to the noisy components of the reward realizations. Note that due to this multi-environment setup, the knowledge gained from all the collected N datasets $\mathcal{Z}_i, i = 1, \ldots, N$ may not transfer well to the new task parameter θ_{N+1} since the learner does not know the environment from which the new task originates. Accordingly, we need an algorithm that first decides to which environment the new task belongs. Then it uses an appropriate meta-learning scheme that considers the differences of the environments in order to leverage the task similarities to minimize the transfer regret. In order to provide training data for the algorithm to be able to distinguish between the environments (i.e., gain information regarding the unknown task distributions ρ_ν), we require that the learner is presented with a number of initial tasks with labeled environments in order to obtain a stationary behaviour in terms of estimating a good bias parameter. Specifically, we introduce the following assumption.

Assumption 1: We assume that for the first N completed tasks, the learner has knowledge regarding the environment from which each task originates. In particular, we assume the learner has access to the sets $\mathcal{S}_\nu = \{i : \theta_i \sim \rho_\nu, i = 1, \dots, N\}$ for $\nu = 1, \dots, m$. We let $N_\nu = |\mathcal{S}_\nu|$.

D. Model Assumptions

Next, we present two more assumptions that are standard in the bandit literature [10].

Assumption 2: For all t, ξ_t are conditionally zero-mean R-sub-Gaussian noise variables, i.e., $\mathbb{E}[\xi_t|\mathcal{F}_{t-1}] = 0$, and $\mathbb{E}[e^{\lambda \xi_t}|\mathcal{F}_{t-1}] \le \exp\left(\frac{\lambda^2 R^2}{2}\right), \forall \lambda \in \mathbb{R}$.

Assumption 3: There exists a positive constant S and L such that for every LB problem, $\|\theta\|_2 \le S$ and $\|x\|_2 \le L$ for every $x \in \cup_{s=1}^{T} \mathcal{D}_s$. Also, $x^\top \theta \in [-1, 1]$, for all $x \in \mathcal{D}_t$.

III. Background on Biased OFUL

Before introducing our proposed MEML-OFUL algorithm for the setting introduced in Section II-C, in the following, we first review the OFUL algorithm and the biased version of OFUL, which our algorithm builds upon.

A. OFUL

For the single LB problem in Section II-A, we consider the OFUL algorithm [10]. At each round $t \in [T]$, the algorithm uses the previous action-observation pairs and obtains a regularized least-square (RLS) estimate of θ as $\hat{\theta}_t = V_t^{-1} \sum_{s=1}^{t-1} y_s x_s$, where $V_t = \lambda I + \sum_{s=1}^{t-1} x_s x_s^\top$. Then, based on $\hat{\theta}_t$, OFUL builds a confidence set $\mathcal{C}_t(\delta) = \{v \in \mathbb{R}^d : \left\|\hat{\theta}_t - v\right\|_{V_t} \le R\sqrt{d \log\left(\frac{1+tL^2/\lambda}{\delta}\right)} + \sqrt{\lambda} S := \beta_t(\delta)\}$ that includes a true reward parameter θ with probability at least $1 - \delta$. Then, it plays an action x_t by solving $x_t = \arg\max_{x \in \mathcal{D}_t} \max_{v \in \mathcal{C}_t} x^\top v$. For this algorithm, [10] proves a high probability regret bound of order $\mathcal{O}(d\sqrt{T} \log(\frac{TL^2}{\delta}))$.

B. Biased OFUL

For a single LB problem, [1] studies the biased version of the OFUL algorithm, called BIAS-OFUL. In particular, given a bias parameter $h \in \mathbb{R}^d$ for the true reward parameter θ, at each round $t \in [T]$, the RLS-estimate $\hat{\theta}_t^h$ such that $\hat{\theta}_t^h = V_t^{-1} \sum_{s=1}^{t} x_s(y_s - x_s^\top h) + h$. Then, given an oracle that computes $\|h - \theta\|_2$ for their algorithm, they show that they can build a confidence region $\mathcal{C}_t^h(\delta) = \{v \in \mathbb{R}^d : \left\|\hat{\theta}_t^h - v\right\|_{V_t} \le R\sqrt{d \log\left(\frac{1+tL^2/\lambda}{\delta}\right)} + \sqrt{\lambda}\, \|h - \theta\|_2\}$ such that $\theta \in \mathcal{C}_t^h$ with probability at least $1 - \delta$. They also adopt the same action selection rule as the one in OFUL, and using the Corollary 19.3 of [40], they provide an upper bound for the expected regret of their algorithm.

Proposition 3.1 (Lem. 1, [1]): Under Assumptions 2, 3, and considering $\lambda \ge 1$, the expected regret of the BIAS-OFUL is bounded as:

$$\mathbb{E}[R(T, \theta_\star)] \le C\sqrt{Td \log(1 + \frac{TL}{\lambda d})} \left(R\sqrt{d \log(T + T^2 L/(\lambda d))} + \sqrt{\lambda}\, \|h - \theta_\star\|_2\right), \tag{6}$$

where $C > 0$ is a universal constant factor.

It can be seen in (6) that having a good bias parameter $h = \theta_\star$ brings a substantial benefit with respect to the regret (as $\lambda \to \infty$, the regret will tend to zero) in comparison to the unbiased case where $h = 0$.

Then, [1] adopts the BIAS-OFUL algorithm for the meta-learning setting described in Section II-B. They adopt from [25], [41], the idea of adding a bias parameter in a sequence of the tasks that share the same environment and apply it to the linear stochastic bandit framework. Specifically. they show that for the meta-learning problem introduced in Section II-B, running BIAS-OFUL with a bias parameter $h = \bar{\theta} := \mathbb{E}_{\theta \sim \rho}[\theta]$ would significantly speed up the process of learning (i.e., lower regret) with respect to the unbiased case. This holds for a family of task-distributions where the second moment is much larger than the variance as formalized below.

Assumption 4: The task-distribution ρ satisfies:

$$\text{Var}_{\bar{\theta}} = \mathbb{E}_{\theta \sim \rho}[\left\|\theta - \bar{\theta}\right\|_2^2] \ll \mathbb{E}_{\theta \sim \rho}[\|\theta\|_2^2] = \text{Var}_0,$$

Overall, they show the following upper bound for the expected transfer regret defined in (3).

Proposition 3.2 (Lem.2, [1]): Let Assumptions 2, 3 hold, and fix $\lambda = \frac{1}{T \text{Var}_h}$. In the case where the tasks share the same environment ρ satisfying Assumption 4, the expected transfer regret of BIAS-OFUL with a bias parameter h is bounded as:

$$\mathbb{E}_{\theta \sim \rho}\left[\mathbb{E}\left[R(T, \theta)\right]\right] \le dC\sqrt{T \log\left(1 + \frac{T^2 L(\mathbb{E}_{\theta \sim \rho}[\|\theta - h\|_2^2])}{d}\right)}$$

They also propose two strategies to estimate the bias parameter h within the meta-learning setting in order to minimize the transfer regret. In particular, they show that if they can estimate the bias parameter h equal to $\bar{\theta}$ with the meta-learning approach, then according to the Assumption 4, they substantially benefit from the task similarity compared to learning each task separately, i.e., compared to choosing $h = 0$.

IV. Multi-Environment Meta-Learning Algorithm (MEML-OFUL)

MEML-OFUL builds on BIAS-OFUL to address the case where the learning tasks can originate from multiple environments as explained in Section II-C. The summary of MEML-OFUL is presented in Algorithm 1. In particular, in order to minimize the transfer regret in (4), we employ the idea of applying a bias parameter for each task-distribution ρ_ν within the meta-learning setting. For brevity, we will state the results for the case where $m = 2$, i.e., there are only two environments. The extension to the case where $m > 2$ is straightforward.

One of the main challenges of the multi-environment meta-learning problem is that when a new task θ_{N+1} is sampled, the learner does not know from which task distribution this

Algorithm 1: MEML-OFUL algorithm for task $N+1$

```
Input: $\lambda > 1$, $T_0$, $T$, datasets of $N$ completed tasks,
Set $\hat{h}^1_{N+1}$ and $\hat{h}^2_{N+1}$ according to (7)
for $t = 1, \dots, T_0$ do
    Randomly choose $x_{N+1,t} \in \mathcal{D}_t$, and observe the
      reward $y_{N+1,t} = x_{N+1,t}^\top \theta_{N+1} + \xi_{N+1,t}$.
    Compute $\hat{\theta}_{N+1,t} = V_{N+1,t}^{-1} \sum_{s=1}^{t-1} y_{N+1,s} x_{N+1,s}$ .
Select the bias
  $\hat{h}_{N+1} = \arg\min_{j \in \{1,2\}} \left\| \hat{\theta}_{N+1,T_0} - \hat{h}^j_{N+1} \right\|_2^2$.
for $t = T_0, \dots, T+1$ do
    Build a confidence region $\mathcal{C}_t^h$ with a bias $\hat{h}^j_{N+1}$
    Play $x_{N+1,t} = \arg\max_{x \in \mathcal{D}_t} \max_{v \in \mathcal{C}_t^h} x^\top v$
    Observe reward $y_{N+1,t} = x_{N+1,t}^\top \theta_i + \xi_{N+1,t}$
    Update $\hat{\theta}_{N+1,t+1} =$
      $(\lambda I + V_{N+1,t+1})^{-1} \sum_{s=1}^{t} x_{N+1,s}(y_{N+1,s} -$
      $x_{N+1,s}^\top \hat{h}_{N+1}) + \hat{h}_{N+1}$
    Update $V_{N+1,t+1} = \sum_{s=1}^{t} x_{N+1,s} x_{N+1,s}^\top$
```

task originated, and hence which bias parameter to apply. In particular, for a new task θ_{N+1}, there exist two bias parameters $\hat{h}^1_N$ and $\hat{h}^2_N$ from previously completed tasks in each environment, which can be used to transfer information to the new task. If the leaner selects the wrong bias parameter, then the regret of the new task could be larger than that of the unbiased case (i.e., independent learning of each task). To handle this issue, MEML-OFUL performs a pure exploration phase for the first T_0 rounds of a new task in order to calculate the RLS-estimate $\hat{\theta}_{N+1,T_0}$ of the new task parameter θ_{N+1}. Then, it chooses the bias parameter that has the smallest square Euclidean distance from $\hat{\theta}_{N+1,T_0}$. It then runs BIAS-OFUL to complete the task and update the bias parameter.

We note that the regret grows linearly with the length of the exploration phase. However, longer exploration allows the learner to compute more accurate RLS-estimates of the new task parameter, and hence minimize the misclassification probability (i.e., selecting the wrong bias parameter), presenting a design trade-off. Note that even with a perfect estimate of θ_{N+1}, misclassification can still happen as θ_{N+1} might have non-zero mass in both distributions ρ_1, ρ_2. As such, we need to carefully design the length of the pure exploration phase to be just long enough in order to compute a good estimate of the task parameter for classification, but not any longer so as to not adversely affect the regret. After restricting the class of distribution ρ_ν, our analysis in Theorem 5.1 shows that we can set the length of the exploration phase such that it is constant with respect to T and it inversely depends on the distance between the expected value ($\mu_i, i = 1, 2$) of the task distributions (i.e., it captures the difference of the two environments).

We emphasize that in the case where the new task parameter θ_{N+1} has a non-zero probability of being sampled from both ρ_1 and ρ_2, there always exists a non-zero probability that MEML-OFUL chooses the wrong bias parameter, and hence suffers a larger regret. Therefore, in order to bound the regret, we need to compute the probability that MEML-OFUL misclassifies the environment. To do so, we make the following assumption on the family of task-distributions we study.

Assumption 5: We assume that for $i = 1, 2$, the task-distribution ρ_i is a multivariate distributions on $\mathbb{R}^d$ such that any sample $x \sim \rho_i$ can be written as $x = \mu_i + z$, where $z \in \mathbb{R}^d$ has i.i.d. zero-mean, K-sub-Gaussian entries with a bounded support in $\mathbb{R}^d$, and $\mu_i \in \mathbb{R}^d$ is a fixed (but otherwise unknown) vector. We define the constant $\gamma = \|\mu_2 - \mu_1\|_2$. Also, we assume $\mathbb{E}_{\theta \sim \rho_i}[\|\theta - \mu_i\|_2^2] \ll \gamma$ for $i = 1, 2$.

Then, we employ the biasing idea proposed in [1], which is based on averaging the RLS-estimate of the task parameters of the first N completed tasks with labeled environment (i.e., $\hat{\theta}_{j,T}$, $j = 1, \dots, N$) without considering any bias . In particular, for a new task $N+1$, we set

$$\hat{h}^1_{N+1} = \frac{1}{N_1} \sum_{j \in \mathcal{S}_1} \hat{\theta}_{j,T}, \quad \hat{h}^2_{N+1} = \frac{1}{N_2} \sum_{j \in \mathcal{S}_2} \hat{\theta}_{j,T}. \tag{7}$$

Then, after the pure exploration phase, the algorithm decides to use $\hat{h}^1_{N+1}$ based on the Euclidean distance from the RLS-estimate of the new taks parameter (or similarly, $\hat{h}^2_{N+1}$).

V. Regret Bound

In this section, we show the following bound on the transfer regret in (5) for MEML-OFUL algorithm.

Theorem 5.1: Let Assumptions 1-5 hold, and let the bias parameters be defined as in (7). Then, for given prior probabilities $\mathbb{P}(\nu = 1) = p_1$, $\mathbb{P}(\nu = 2) = p_2$, the transfer regret defined in (5) is upper bounded as follows

$$\mathcal{R}(T) \leq \sum_{i=1}^{2} p_i (2T_0 +$$
$$dC \sqrt{(T - T_0) \log \left(1 + \frac{(T-T_0)^2 L \left(\mathrm{Var}^i_{\mu_i} + \tau^i_N\right)}{d}\right)})$$

where $T_0 > \mathcal{O}\left(\frac{R\sqrt{d \log(4/\delta)}}{\gamma - \sqrt[4]{K^2 d(\frac{1}{N_1} + \frac{1}{N_2})} - 2K\sqrt{\log(4/\delta)}}\right)$ for large enough N_1, N_2 such that $\gamma^2 \geq K\sqrt{d(\frac{1}{N_1} + \frac{1}{N_2})}$, and $\mathrm{Var}^i_{\mu_i} = \mathbb{E}_{\theta \sim \rho_i}[\|\theta - \mu_i\|_2^2]$. Moreover, we have with probability at least $1 - \delta$ for $\delta \in (\frac{1}{2e^{\frac{\gamma^2}{4K^2}}}, 1)$, for $i = 1, 2$:

$$\sqrt{\tau^i_N} = \sqrt{\left\|\mu_i - \hat{h}_N\right\|} \leq \mathcal{O}\Big(\frac{2S \log(2/\delta)\sqrt{\mathbb{E}_{\theta \sim \rho_i}[\|\theta\|_2^2]}}{N_i}$$
$$+ \max_{j \in \mathcal{S}_i}\{\frac{\beta_{j,T}(1/T)}{\sqrt{\lambda + \lambda_{\min}(V_{j,T})}}\} + \delta\left(\gamma + \frac{2S}{N}\right)\Big). \tag{8}$$

The last term in RHS of (8) comes from the misclassification of the environment for the new task. In particular, as N grows to infinity, the RHS of (8) is dominated by $\max_{j \in \mathcal{S}_i}\{\frac{\beta_{j,T}(1/T)}{\sqrt{\lambda + \lambda_{\min}(V_{j,T})}}\} + \delta\gamma$, where the first term comes from the variance of the environment ρ_i and the second term is caused by the inevitable effect of misclassification of the environment. We note that the lower bound for the number of

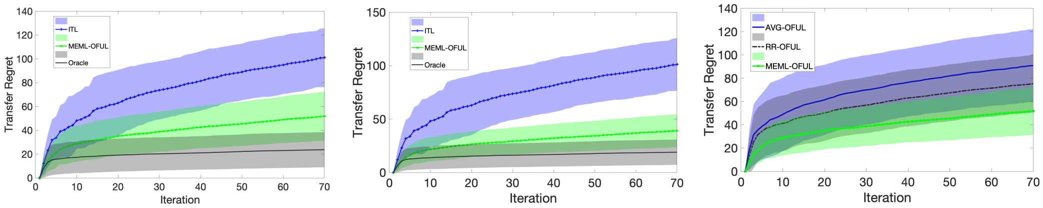

Fig. 1. Transfer regret of MEML-OFUL versus ITL and Oracle algorithm measured after $N_1 = 10$, $N_2 = 10$ training tasks. Left: $\lambda = 1$; Middle: $\lambda = 200$; Right: Comparing the transfer regret of our algorithm versus AVG-OFUL and RR-OFUL algorithms from [1] for the mixture distribution with $\lambda = 1$.

rounds in the pure exploration phase is constant with respect to the time T, and it inversely depends on the probability that MEML-OFUL misclassifies the environment. We have also shown a lower bound on the number of tasks with a labeled environment that our algorithm requires in order to reach a stationary behaviour in estimating the bias.

VI. Discussion

A. When it is beneficial to apply MEML-OFUL?

Defining a mixture distribution $\rho = p_1\rho_1 + p_2\rho_2$ with known mixing probabilities such that $p_1 + p_2 = 1$, one may ask the question: what happens if we apply the algorithm in [1] to our multi-environment meta-learning setting? We answer this question in several steps.

Consider the BIAS-OFUL algorithm in [1] applied to a new single-environment task-distribution $\rho = p_1\rho_1 + p_2\rho_2$ with expected value $\mu = p_1\mu_1 + p_2\mu_2$. In order to leverage the task similarities, BIAS-OFUL requires this mixture task-distribution ρ to satisfy Assumption 4, i.e., that the task-distribution has a non-zero expected value. Therefore, for any family of environments such that $\mu = p_1\mu_1 + p_2\mu_2 = 0$ (e.g., $\mu_1 = -\mu_2$ and $p_1 = p_2 = 1/2$), BIAS-OFUL would not necessarily out-perform independent learning. However, MEML-OFUL will not encounter this problem, since it interacts with each environment separately, and as long as $\mu_1, \mu_2 \neq 0$, it leverages the task similarities of each environment to bring substantial benefit with respect to the unbiased case.

Next consider multi-environment settings with $\mu = p_1\mu_1 + p_2\mu_2 \neq 0$. What do we gain from adopting the MEML-OFUL algorithm over applying the BIAS-OFUL algorithm to the mixture distribution ρ? To start at a high level, we consider the case that the meta-learning algorithm perfectly estimates the bias parameter to be equal to the expected value of the environment from which the task is sampled. In this case, Proposition 3.2 shows the following bound on the transfer-regret of the mixture distribution for BIAS-OFUL:

$$dC\sqrt{T\log\left(1+\frac{T^2L\left(p_1\mathrm{Var}^1_{\mu_1}+p_2\mathrm{Var}^2_{\mu_2}\right)}{d}\right)}, \tag{9}$$

For the same case, from the result of Theorem 5.1, we obtain the following bound on the transfer-regret of the MEML-OFUL algorithm:

$$\mathcal{R}(T) \leq p_1(2T_0 + dC\sqrt{(T-T_0)\log\left(1+\frac{T^2L\mathrm{Var}^1_{\mu_1}}{d}\right)})$$
$$+ p_2(2T_0 + dC\sqrt{(T-T_0)\log\left(1+\frac{T^2L\mathrm{Var}^2_{\mu_2}}{d}\right)}). \tag{10}$$

Now, we know from Theorem 5.1 that T_0 is constant with respect to the time horizon T, and since the log and square root functions are strictly concave, we can conclude from Jensen's inequality that for a large enough T, the regret bound in (10) is less than the one in (9). This shows that it cannot hurt to adopt the MEML-OFUL over the BIAS-OFUL algorithm in the multi-environment settings *if* the number of training tasks N is sufficiently large to provide a close to exact estimate of expected value for each environment. That being said, adopting MEML-OFUL has several extra requirements: 1) MEML-OFUL requires at least $N_1 + N_2$ training datasets of labeled environment such that $\gamma^2 \geq K\sqrt{d(\frac{1}{N_1}+\frac{1}{N_2})}$ as stated in Theorem 5.1. Of course, just meeting this minimum training requirement does not guarantee that MEML-OFUL would outperform BIAS-OFUL; 2) MEML-OFUL requires a pure exploration phase to estimate the task environment, during which it incurs linear regret; 3) MEML-OFUL requires knowledge of at least a lower bound on $\gamma = \|\mu_2 - \mu_1\|_2$, since the length of the pure exploration phase depends on it.

VII. Numerical Results

We investigate numerically the effectiveness of MEML-OFUL for the meta-learning setting in Section II-C on synthetic data. As mentioned in Section III, in order to build the confidence regions, the algorithm requires the value $\left\|\theta_i - \hat{h}_i\right\|_2$ which we upper bound similar to [1]. We first generate two environments in agreement with Assumption 4 such that ρ_1 and ρ_2 are Gaussian distributions with means $\mu_1 = [1;1]$ and $\mu_2 = [3;3]$ and $\mathrm{Var}^i_{\mu_i} = 1,\ i = 1, 2$. In all the implementations, we used $T = 70, \delta = 1/T, R = 0.1$. The transfer-regret figures are averaged over $N = 10$ test tasks, where each environment ρ_i was sampled with probability $p_i = 1/2$. For the decision set $\mathcal{D}$, we follow a similar approach to the one in [1]. In Figure 1 the shaded regions show standard deviation around the mean. We plot the MEML-OFUL algorithm as well as the independent task learning (ITL) policy, which completes each learning task separately. Also, we plot the Oracle policy that knows the mean of each environment μ_1 and μ_2. The transfer regret shown in Fig. 1 (left) are for $\lambda = 1$ and Fig. 1 are for $\lambda = 200$. Moreover, we adopt the two proposed algorithm AVG-OFUL and RR-OFUL proposed in [1] to our setting in the mixture environment $\rho = (\rho_1 + \rho_2)/2$ which is a Gaussian distribution with mean $\mu = [2; 2]$. Figure 1 (right) shows that our proposed algorithms out-perform both algorithms in [1], which supports our discussion in Section VI-A.

References

[1] L. Cella, A. Lazaric, and M. Pontil, "Meta-learning with stochastic linear bandits," *arXiv preprint arXiv:2005.08531*, 2020.

[2] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2-3, pp. 235–256, May 2002. [Online]. Available: https://doi.org/10.1023/A:1013689704352

[3] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 661–670.

[4] S. Filippi, O. Cappe, A. Garivier, and C. Szepesvári, "Parametric bandits: The generalized linear case," in *Advances in Neural Information Processing Systems*, 2010, pp. 586–594.

[5] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933.

[6] E. Kaufmann, N. Korda, and R. Munos, "Thompson sampling: An asymptotically optimal finite-time analysis," in *International Conference on Algorithmic Learning Theory*. Springer, 2012, pp. 199–213.

[7] D. Russo and B. Van Roy, "Learning to optimize via posterior sampling," *Mathematics of Operations Research*, vol. 39, no. 4, pp. 1221–1243, 2014.

[8] V. Dani, T. P. Hayes, and S. M. Kakade, "Stochastic linear optimization under bandit feedback," *21st Annual Conference on Learning Theory*, 2008.

[9] P. Rusmevichientong and J. N. Tsitsiklis, "Linearly parameterized bandits," *Mathematics of Operations Research*, vol. 35, no. 2, pp. 395–411, 2010.

[10] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," in *Advances in Neural Information Processing Systems*, 2011, pp. 2312–2320.

[11] A. Moradipari, C. Thrampoulidis, and M. Alizadeh, "Stage-wise conservative linear bandits," *Advances in neural information processing systems*, vol. 33, pp. 11 191–11 201, 2020.

[12] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," in *International Conference on Machine Learning*, 2013, pp. 127–135.

[13] M. Abeille, A. Lazaric *et al.*, "Linear thompson sampling revisited," *Electronic Journal of Statistics*, vol. 11, no. 2, pp. 5165–5197, 2017.

[14] A. Moradipari, S. Amani, M. Alizadeh, and C. Thrampoulidis, "Safe linear thompson sampling with side information," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3755–3767, 2021.

[15] A. Moradipari, M. Alizadeh, and C. Thrampoulidis, "Linear thompson sampling under unknown linear constraints," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3392–3396.

[16] R. K. Ando, T. Zhang, and P. Bartlett, "A framework for learning predictive structures from multiple tasks and unlabeled data." *Journal of Machine Learning Research*, vol. 6, no. 11, 2005.

[17] M. Pontil and A. Maurer, "Excess risk bounds for multitask learning with trace norm regularization," in *Conference on Learning Theory*. PMLR, 2013, pp. 55–76.

[18] A. Maurer, M. Pontil, and B. Romera-Paredes, "Sparse coding for multitask and transfer learning," in *International conference on machine learning*. PMLR, 2013, pp. 343–351.

[19] ——, "The benefit of multitask representation learning," *Journal of Machine Learning Research*, vol. 17, no. 81, pp. 1–32, 2016.

[20] G. Cavallanti, N. Cesa-Bianchi, and C. Gentile, "Linear algorithms for online multitask classification," *The Journal of Machine Learning Research*, vol. 11, pp. 2901–2934, 2010.

[21] S. S. Du, W. Hu, S. M. Kakade, J. D. Lee, and Q. Lei, "Few-shot learning via learning the representation, provably," *arXiv preprint arXiv:2002.09434*, 2020.

[22] J. Baxter, "A model of inductive bias learning," *Journal of artificial intelligence research*, vol. 12, pp. 149–198, 2000.

[23] P. Alquier, M. Pontil *et al.*, "Regret bounds for lifelong learning," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 261–269.

[24] T. Schaul and J. Schmidhuber, "Metalearning," *Scholarpedia*, vol. 5, no. 6, p. 4650, 2010.

[25] G. Denevi, C. Ciliberto, R. Grazzi, and M. Pontil, "Learning-to-learn stochastic gradient descent with biased regularization," in *International Conference on Machine Learning*, 2019, pp. 1566–1575.

[26] C. Finn, A. Rajeswaran, S. Kakade, and S. Levine, “Online meta-learning,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 1920–1930.

[27] M. Khodak, M.-F. Balcan, and A. Talwalkar, “Adaptive gradient-based meta-learning methods,” *arXiv preprint arXiv:1906.02717*, 2019.

[28] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, “Meta-learning with differentiable convex optimization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 657–10 665.

[29] D. Calandriello, A. Lazaric, and M. Restelli, “Sparse multi-task reinforcement learning,” *Intelligenza Artificiale*, vol. 9, no. 1, pp. 5–20, 2015.

[30] M. G. Azar, A. Lazaric, and E. Brunskill, “Sequential transfer in multi-armed bandit with finite set of models,” *arXiv preprint arXiv:1307.6887*, 2013.

[31] J. Zhang and E. Bareinboim, “Transfer learning in multi-armed bandit: a causal approach,” in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, 2017, pp. 1778–1780.

[32] A. A. Deshmukh, U. Dogan, and C. Scott, “Multi-task learning for contextual bandits,” *arXiv preprint arXiv:1705.08618*, 2017.

[33] M. Soare, O. Alsharif, A. Lazaric, and J. Pineau, “Multi-task linear bandits,” in *NIPS2014 Workshop on Transfer and Multi-task Learning: Theory meets Practice*, 2014.

[34] B. Liu, Y. Wei, Y. Zhang, Z. Yan, and Q. Yang, “Transferable contextual bandit for cross-domain recommendation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[35] J. Yang, W. Hu, J. D. Lee, and S. S. Du, “Provable benefits of representation learning in linear bandits,” *arXiv preprint arXiv:2010.06531*, 2020.

[36] A. Moradipari, Y. Abbasi-Yadkori, M. Alizadeh, and M. Ghavamzadeh, “Parameter and feature selection in stochastic linear bandits,” *arXiv preprint arXiv:2106.05378*, 2021.

[37] J. Hong, B. Kveton, M. Zaheer, and M. Ghavamzadeh, “Hierarchical bayesian bandits,” *arXiv preprint arXiv:2111.06929*, 2021.

[38] B. Kveton, M. Konobeev, M. Zaheer, C.-w. Hsu, M. Mladenov, C. Boutilier, and C. Szepesvari, “Meta-thompson sampling,” *arXiv preprint arXiv:2102.06129*, 2021.

[39] J. Hong, B. Kveton, M. Zaheer, M. Ghavamzadeh, and C. Boutilier, “Thompson sampling with a mixture prior,” *arXiv preprint arXiv:2106.05608*, 2021.

[40] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

[41] G. Denevi, C. Ciliberto, D. Stamos, and M. Pontil, “Learning to learn around a common mean,” *Advances in Neural Information Processing Systems*, vol. 31, pp. 10 169–10 179, 2018.