Meta-learning with Stochastic Linear Bandits

Leonardo Cella 1 2  Alessandro Lazaric 3  Massimiliano Pontil 2

Abstract
We investigate meta-learning procedures in the setting of stochastic linear bandits tasks. The goal is to select a learning algorithm which works well on average over a class of bandits tasks, that are sampled from a task-distribution. Inspired by recent work on learning-to-learn linear regression, we consider a class of bandit algorithms that implement a regularized version of the well-known OFUL algorithm, where the regularization is a square euclidean distance to a bias vector. We first study the benefit of the biased OFUL algorithm in terms of regret minimization. We then propose two strategies to estimate the bias within the learning-to-learn setting. We show both theoretically and experimentally, that when the number of tasks grows and the variance of the task-distribution is small, our strategies have a significant advantage over learning the tasks in isolation.

1. Introduction
The multi-armed bandit MAB (Lattimore & Szepesvári, 2020; Auer et al., 2002; Siegmund, 2003; Robbins, 1952; Cesa-Bianchi, 2016; Bubeck et al., 2012) is a simple framework formalizing the online learning problem constrained to partial feedback. In the last decades it has receiving increasing attention due to its wide practical importance and the theoretical challenges in designing principled and efficient learning algorithms. In particular, applications range from recommender systems (Li et al., 2010; Cella & Cesa-Bianchi, 2019; Bogers, 2010), to clinical trials (Villar et al., 2015), and to adaptive routing (Awerbuch & Kleinberg, 2008), among others.

In this paper, we are concerned with linear bandits (Abbasi-Yadkori et al., 2011; Chu et al., 2011; Auer, 2003), a consolidated MAB setting in which each arm is associated with a vector of features and the arm payoff function is modeled by a (unknown) linear regression of the arm feature vector. Our study builds upon the OFUL algorithm introduced in (Abbasi-Yadkori et al., 2011), which in turned improved the theoretical analysis initially investigated in (Chu et al., 2011; Auer, 2003). Nonetheless, it may still require a long exploration in order to estimate well the unknown linear regression vector. An appealing approach to solve this bottleneck is to leverage already completed tasks by transferring the previously collected experience to speedup the learning process. This framework finds its most common application in the recommendation system domain, where we wish to recommend contents to a new user by matching his preference. Our objective is to rely on past interactions corresponding to navigation of different users to speedup the learning process.

Previous Work During the past decade, there have been numerous theoretical investigation of transfer learning, with a particular attention to the problems of multi-task (MTL) (Ando & Zhang, 2005; Maurer & Pontil, 2013; Maurer et al., 2013; 2016; Cavallanti et al., 2010) and learning-to-learn (LTL) or meta-learning (Baxter, 2000; Alquier et al., 2017; Denevi et al., 2018a;b; 2019; Pentina & Urner, 2016). The main difference between these two settings is that MTL aims to solve the problem of learning well on a prescribed set of tasks (the learned model is tested on the same tasks used during training), whereas LTL studies the problem of selecting a learning algorithm that works well on tasks from a common environment (i.e. sampled from a prescribed distribution), relying on already completed tasks from the same environment (Pentina & Urner, 2016; Balcan et al., 2019; Denevi et al., 2018a; 2019). In either case the base tasks considered have always been supervised learning ones. Recently, the MTL setting has been extended to a class of bandit tasks, with encouraging results empirically and theoretically (Azar et al., 2013; Calandriello et al., 2014; Zhang & Bareinboim, 2017; Deshmukh et al., 2017; Liu et al., 2018), as well as the case where tasks belong to a (social) graph, a setting that is usually referred to as collaborative linear bandit (Cesa-Bianchi et al., 2013; Soare et al., 2014; Gentile et al., 2014; 2017). Differently from these works, the principal goal of this paper is to investigate the adoption of the meta-learning framework, which has been
Contributions} Our contribution is threefold. First, we introduce in Section 3 a variant of the OFUL algorithm in which the regularization term is modified by introducing a bias vector, analyzing the impact of the bias in terms of regret minimization. Second, and more importantly, in Sections 4 and 5 we propose two alternative approaches to estimate the bias, within the meta-learning setting. We establish theoretical results on the regret of these methods, highlighting that, when the task-distribution has a small variance and the number of tasks grows, adopting the proposed meta-learning methods lead a substantial benefit in comparison to using the standard OFUL algorithm. Finally, in Section 6 we compare experimentally the proposed methods with respect to the standard OFUL algorithm on both synthetic and real data.

2. Learning Foundations

In this section we start by briefly recalling the standard stochastic linear bandit framework and we then present the considered LTL setting.

2.1. Linear Stochastic Bandits

Let $T$ be a positive integers and let $[T] = \{1, \ldots, T\}$. A Linear Stochastic MAB is defined by a sequence of $T$ interactions between the agent and the environment. At each round $t \in [T]$, the learner is given a decision set $D_t \subseteq \mathbb{R}^d$ from which it has to pick an arm $x_t \in D_t$. Subsequently, it observes the corresponding reward $y_t = x_t^\top w^* + \eta_t$ which is defined by a linear relation with respect to an unknown parameter $w^* \in \mathbb{R}^d$ combined with a sub-gaussian random noise term $\eta_t$. Thanks to the knowledge of the true parameter $w^*$, at each round $t$ the optimal policy picks the arm $x_t^* = \arg \max_{x \in D_t} x^\top w^*$, maximizing the instantaneous reward. The learning objective is to maximize the cumulative reward, or equivalently, to minimize the pseudo-regret

$$R(T, w^*) = \sum_{t=1}^{T} (x_t^* - x_t)^\top w^*.$$

As learning algorithm we consider OFUL (Abbasi-Yadkori et al., 2011). At each round $t \in [T]$, it estimates $w^*$ by ridge-regression over the observed arm reward pairs, that is,

$$\tilde{w}_{t+1} = \arg \min_{w \in \mathbb{R}^d} \norm{X_t w - y_t}^2 + \lambda \norm{w}^2 \tag{1}$$

where $X_t$ is the matrix whose rows are $x_1^\top, \ldots, x_T^\top$, $I$ is the $d \times d$ identity matrix and $y_t = (y_1, \ldots, y_T)^\top$. A key insight behind OFUL is to update online a confidence interval $C_t$ containing the true parameter $w^*$ with high probability and centered in $\tilde{w}_t^\top$. According to Theorem 2 of (Abbasi-Yadkori et al., 2011), assuming $\norm{w^*}_2 \leq S$ and $\norm{x}_2 \leq L$

$$\forall x \in \cup_{s=1}^{T} D_s$$

then for any $\delta > 0$, w.p. $1 - \delta$, for all $t \geq 0$, $w^*$ lies in

$$C_t(\delta) = \left\{ w \in \mathbb{R}^d : \norm{\tilde{w}_t^\top - w} \|V_t^\top \leq \left( \frac{d \log \frac{1 + TL^2/\delta}{\delta} + \lambda^2 S}{\delta} \right) \right\} \tag{2}$$

where $V_t^\top = \lambda I + X_t^\top X_t$. According to the optimism in the face of uncertainty principle, at each round $t$ OFUL picks the arm $x_t$ by solving the following optimization problem:

$$x_t = \arg \max_{x \in D_t} \max_{\tilde{w}_t^\top \in C_t} x^\top \tilde{w}_t. \tag{3}$$

As was proved in Lemma 5 of (Kuzborskij et al., 2019), this corresponds to pick:

$$x_t \in \arg \max_{x \in D_t} \left\{ x^\top \tilde{w}_{t-1} + \beta_{t-1}(\delta) \norm{x} \|V_{t-1}^\top \right\} \tag{4}$$

Finally, with probability at least $1 - \delta$, OFUL satisfies (see Theorem 3 of (Abbasi-Yadkori et al., 2011)):

$$R(T, w^*) \leq 4\sqrt{T d \log \left(1 + \frac{TL}{\lambda d}\right) \left( \lambda^2 S + \frac{2 \log(1/\delta)}{\delta} \right)} + \frac{2 \log(1/\delta)}{\delta}.$$

We can now formally introduce the considered LTL learning framework for the family of tasks we analyze in this work: biased regularized linear stochastic bandits.

2.2. LTL with Linear Stochastic Bandits

We assume that each learning task $w \in \mathbb{R}^d$ representing a linear bandit, is sampled from a task-distribution $\rho$ of bounded support in $\mathbb{R}^d$. The objective is to design a meta-learning algorithm which is well suited to the environment. Specifically, we assume to receive a sequence of tasks $w_1, \ldots, w_N, \ldots$ which are independently sampled from the task-distribution (environment) $\rho$. Due to the interactive nature of the bandit setting, we do not have any prior information related to a new task; we collect information about it along the interaction with the environment. After completing the $j$-th task, we store the whole interaction in a dataset $Z_j$ which is formed by $T$ entries $(x_{j,t}, y_{j,t})_{t=1}^T$. Clearly, the dataset entries are not i.i.d sampled from a given distribution, but each dataset $Z_j$ corresponds to the recording of the learning policy in terms of the arm $x_{j,t}$ picked from the decision set $D_t$ and its corresponding reward $y_{j,t}$ while facing the task specified by the unknown vector $w_j$.

Starting from these datasets, we wish to design an algorithm $\mathcal{A}$ which suffers a low regret on a new task $w_{N+1} \sim \rho$. This can be stated into requiring that $\mathcal{A}$ trained over $N$ datasets has small transfer-regret:

$$\mathcal{R}(T, \rho) = \mathbb{E}_{w \sim \rho} \left[ \mathbb{E}[R(T, w)] \right]$$
where the inner expectation is with respect to rewards realizations due to their noisy components.

3. Biased Regularized OFUL

We now introduce BIAS-OFUL, a biased version of OFUL, which is instrumental for our meta-learning setting. Although not feasible, the proposed algorithm it serves as a basis to study the theoretical properties of meta-learning with stochastic linear bandit tasks. In Section 6 we will present a more practical version of it.

Regularized Confidence Sets The idea of following a bias in a specific family of learning algorithms is not new in the ITL literature (Denevi et al., 2018a; 2019; 2018b). Inspired by (Denevi et al., 2019) we modify the regularization in the computation of the confidence set centroid \( \hat{w}_t^h \), where the regularization is now defined as a square euclidean distance to the bias parameter \( h \in \mathbb{R}^d \). Given a fixed vector \( h \), at each round \( t \in [T] \) BIAS-OFUL estimates the regularized centroid of the confidence ellipsoid as

\[
\hat{w}_t^h = \arg \min_w \| X_t^\top w - Y_t \|_2^2 + \lambda \| w - h \|_2^2
\]

whose solution is given by

\[
\hat{w}_t^h = (V_t^\lambda)^{-1} X_t^\top (Y_t - X_t h) + h. \tag{5}
\]

This result follows directly from the standard ridge-regression by using the substitution \( v = w - h \).

As we have mentioned in the previous section, at each round \( t \) OFUL keeps also updated a confidence interval \( C_t \) (see Equation 2) centered in \( \hat{w}_t^h \) which contains \( w^* \) with high probability. We now derive a confidence set for the biased regularized estimate \( \hat{w}_t^h \), assuming that we have access to an oracle to compute the distance \( \| h - w^* \|_2 \). This seems quite restrictive, however later in the paper we will show how leveraging similar related tasks we can exploit this bound to take advantage of the bias version of OFUL, without having to know the above distance a-priori.

Theorem 1. Assuming \( \| h \|_2 \leq S, \| w^* \|_2 \leq S \) and \( \| x \|_2 \leq L \) \( \forall x \in \cup_{s=1}^d D_s \), then for any \( \delta > 0 \), with probability at least \( 1 - \delta \), \( \forall t \geq 0 \), \( w^* \) lies in the set

\[
C_t^h(\delta) = \left\{ w \in \mathbb{R}^d : \| \hat{w}_t^h - w \|_{V_t^\lambda} \leq \lambda^{\frac{1}{2}} \| h - w^* \|_2 + R \sqrt{2 \log \left( \frac{\det (V_t^\lambda)^{1/2}}{\det (\lambda I)^{1/2} \delta} \right)} = \beta_t^h(\delta) \right\}, \tag{6}
\]

The proof can be found in the appendix material. We will now study the impact of the bias \( h \) in terms of regret.

3.1. Regret Analysis with Fixed Bias

Given the confidence set defined in Theorem 1 and the optimism principle translated into selecting the next arm according to Equation 4, we can analyze the expected pseudo-regret depending on the value of \( h \).

Lemma 1. (BIAS-OFUL Expected Regret) Under the same assumptions of Theorem 1, if in addition, for all \( t \) and all \( x \in D_t \), \( x^\top w^* \in [-1, 1] \), and considering \( \lambda \geq 1 \), we have

\[
\bar{R}(T, w^*) = E [R(T, w^*)] \leq C \sqrt{T d \log \left( 1 + \frac{TL}{\lambda d} \right)} \left( \lambda^{\frac{1}{2}} S + R \sqrt{d \log(T + T^2 L/(\lambda d))} \right)
\]

where the expectation is respect to the reward generation and \( C > 0 \) is a constant factor.

We now analyze the regret for two different values of \( h \). In particular we wish to highlight how setting a good bias can speedup the process of learning with respect to using the standard OFUL approach (Abbasi-Yadkori et al., 2011).

Corollary 1. Under the conditions of Lemma 1, the following bounds on the expected regret of BIAS-OFUL holds:

(i) Independent Task Learning (ITL), given by setting \( h = 0 \) satisfies the following expected regret bound

\[
\bar{R}(T, w^*) \leq C \sqrt{T d \log \left( 1 + \frac{TL}{\lambda d} \right)} \left( \lambda^{\frac{1}{2}} S + R \sqrt{d \log(T + T^2 L/(\lambda d))} \right)
\]

which is of order \( O(d \sqrt{T}) \) for any \( \lambda \geq 1 \).

(ii) The Oracle, given by setting \( h = w^* \) satisfies

\[
\bar{R}(T, w^*) \leq C \sqrt{T d \log \left( 1 + \frac{TL}{\lambda d} \right)} \left( R \sqrt{d \log(T + T^2 L/(\lambda d))} \right)
\]

which is 0 as \( \lambda \to \infty \).

The proofs can be found in the supplementary material. The main intuition is that, as long as we can set \( h = w^* \), the bigger the the regularization parameter \( \lambda \) is, the more the Oracle policy tends to select the arm only based on \( w^* \), thereby becoming equivalent to the optimal policy.
3.2. Transfer Regret Analysis with Fixed Bias

Following the above analysis for the single task case, we now study the impact of the bias in the transfer regret bound. To this end, we introduce the variance and the mean absolute distance of a bias vector $h$ relative to the environment of task,

$$\text{Var}_h = \mathbb{E}_{w \sim \rho}[\|w - h\|^2], \quad \text{Mar}_h = \mathbb{E}_{w \sim \rho}[\|w - h\|]$$

and we observe that $w = \mathbb{E}_{w \sim \rho}w = \arg\min_{h \in \mathbb{R}^d}\text{Var}_h$ and $m = \arg\min_{h \in \mathbb{R}^d}\text{Mar}_h$. With this in hand, we can now analyze how the transfer regret can be upper bounded as a function of the introduced terms.

**Lemma 2.** (Transfer Regret Bound) Under the same conditions in Theorem 1 and Lemma 1, the expected transfer regret of BIAS-OFUL can be upper bounded as:

$$\mathcal{R}(T, \rho) \leq C \sqrt{Td \lambda \log \left(1 + \frac{TL}{\lambda d}\right)} \text{Mar}_h +$$

$$+ RCd \sqrt{T \log \left(1 + \frac{T^2L}{\lambda d}\right)} \log \left(1 + \frac{TL}{\lambda d}\right)$$

$$\leq C \sqrt{Td \lambda \log \left(1 + \frac{TL}{\lambda d}\right)} \text{Var}_h +$$

$$+ RCd \sqrt{T \log \left(1 + \frac{T^2L}{\lambda d}\right)} \log \left(1 + \frac{TL}{\lambda d}\right)$$

**Proof.** The first statement is the expectation with respect to the task-distribution $\rho$ applied to Lemma 1, while the second follows by applying Jensen’s inequality. □

We can now replicate what we have done in Corollary 1 and consider the transfer regret bound for two different values of the hyper-parameter $h$. The main difference is that here, there is not an a-priori correct value for $h$ as it depends on the task-distribution $\rho$.

**Corollary 2.** Under the same assumptions in Theorem 1 and Lemma 1, and setting $\lambda = \frac{1}{\text{Var}_h}$, the following bounds on the transfer regret hold

(i) Independent Task Learning (ITL), given by setting the bias hyperparameter $h$ equal to 0, satisfies

$$\mathcal{R}(T, \rho) \leq \left[1 + \sqrt{Td \log \left(1 + \frac{T^3 \text{Var}_\rho}{d}\right)}\right]$$

(ii) The Oracle, given by setting the bias hyperparameter $h$ equal to the mean task $w$, satisfies

$$\mathcal{R}(T, \rho) \leq \left[1 + \sqrt{Td \log \left(1 + \frac{T^3 \text{Var}_\rho}{d}\right)}\right]$$

Therefore, running BIAS-OFUL with bias $h$ equal to $w$ brings a substantial benefit with respect to the unbiased case when the second moment of the task-distribution $\rho$ is much bigger than its variance. Specifically, we introduce the following assumption.

**Assumption 1.** (Low Biased Variance)

$$\text{Var}_\rho = \mathbb{E}_{w \sim \rho}[w - w|^2] \ll \mathbb{E}_{w \sim \rho}[w|^2] = \text{Var}_w.$$  \hfill (7)

Notice also that the choice $\lambda = 1/(T \text{Var}_h)$, implies that, as $\text{Var}_w$ tends to 0, the regret upper bound of the oracle case tends to zero too reflecting the result of Corollary 1. More in general, we can state that when the environment (i.e. the task-distribution $\rho$) satisfies Assumption 1, leveraging on tasks similarity would give a substantial benefit compared to learning each task separately. Since in practice the mean task parameter $w$ is unknown, in the following sections we propose two alternative approaches to estimate $w$.

4. A High Variance Solution

In this section, we present our first meta-learning method. We begin by introducing some additional notation. We
let $x_{h,t}^j$ be the arm pulled by the BIAS-OFUL algorithm (Algorithm 1) at round $t$-th of the $j$-th task. We denote by $V_{j,T} = \sum_{s=1}^{T} x_{h,s}^j x_{h,s}^j$ the design matrix computed with the $T$ arms picked during the $j$-th task. For each terminated task $j \in [N]$ we also define $b_{j,T} = X_{j,T} Y_{j,T}$. Finally, we introduce the mean estimation error

$$\epsilon_{N,t}(\rho) = \left\| w - \hat{\lambda}_{N,t} \right\|_2^2,$$

which is the error of our estimate $\hat{\lambda}_{N,t}$ with respect to the true mean task $w$, at round $t$ of the $N+1$-th task.

### 4.1. Averaging the Estimated Task Parameters

An intuitive solution to bound the estimation error $\epsilon_{N,t}$ is to simply average of the estimated task parameters $\hat{\lambda}_{j}$ computed according to Equation 1 on the dataset $Z_j$ without considering any bias.

$$\hat{\lambda}_{N,t+1} = \frac{1}{NT+t} \left( \sum_{j=1}^{N} T \hat{\lambda}_{j} + t \hat{\lambda}_{N+1,t} \right). \quad (8)$$

By adopting this approach, we have the following bound on the transfer regret.

**Theorem 2.** (Transfer Regret Bound). Let the assumptions of Lemma 2 hold and let $\hat{\lambda}_{N,t}$ be defined as in Equation (8). Then, it holds that

$$R(T, \rho) \leq dC \sqrt{\frac{T^2L}{d} \left[ \text{Var}_w + \epsilon_{N,T}(\rho) \right] \log \left( 1 + \frac{\text{Var}_w}{d} \right)}$$

where the mean estimation error can be bounded as

$$\epsilon_{N,T}(\rho) \leq H_\rho(N+1, w) + \max_{1 \leq j \leq N+1} \beta_j \frac{1}{T} \lambda_{\min}^{1/2}(V_{j,T}).$$

Here, $\beta_j \frac{1}{T}$ refers to the confidence interval computed with OFUL (see Equation 2) and $H_\rho(N+1, w) = \left\| w - \hat{\lambda}_{N,T} \right\|_2$ with $\hat{\lambda}_{N,t+1} = \frac{1}{NT+t} \left( \sum_{j=1}^{N} T \hat{\lambda}_{j} + t \hat{\lambda}_{N+1,t} \right)$.

**Proof.** We follow the reasoning in Corollary 2, this time setting $\hat{\lambda} = \hat{\lambda}_{N,T}$, and then observe that

$$\epsilon_{N,T}(\rho) \leq H_\rho(N+1, w) + \max_{1 \leq j \leq N+1} \beta_j \frac{1}{T} \lambda_{\min}^{1/2}(V_{j,T}).$$

The above lemma says that the error $H_\rho(N, w)$ goes to zero as $N$ grows to infinity. Therefore the estimation error $\epsilon_{N,t}(\rho)$ is dominated by the “variance” term $\max_{1 \leq j \leq N+1} \beta_j \frac{1}{T} \lambda_{\min}^{1/2}(V_{j,T})$, associated with the worst past task. By relying on linear regression results (Lai & Wei, 1982) we have that $\lambda_{\min}(V_j) \geq T$. Moreover, as $\lambda_{\min}(V_j) \geq \lambda + \lambda_{\min}(V_j)$, we observe an increasing sensitivity of the incurred variance to the $\lambda$ parameter for small value of $T$. Finally, according to our choice of $\lambda = 1/T \text{Var}_w$, the suffered variance increases with the variance of our estimator. The latter in turns increases with the variance of the distribution $\rho$, which corresponds to the case in which Assumption 1 tends to be violated.

### 5. A High Bias Solution

In this section we will present an alternative estimator of the true mean $w$, which is inspired by the existing multi-task bandit literature (Gentile et al., 2014; 2017; Soare et al., 2014). This estimator exploits together all the samples associated to past tasks $Z_1, \ldots, Z_N$, with the aim of reducing the variance. This is unlike the previous estimator which separately considers the ridge-regression estimates $\hat{\lambda}_j$, $\hat{\lambda}_N$ in Equation 8. As we will see, this approach will reduce the variance but it will introduce an extra-bias. Before presenting this second approach we require some more notation. We let $\hat{V}_{N,t} = \sum_{j=1}^{N} V_{N,T} + V_{N+1,t}$ the global design matrix containing the design matrices associated to past tasks $V_{1,T}, \ldots, V_{N,T}$ and the current design matrix $V_{N+1,t}$. Analogously $\hat{b}_{N,t} = \sum_{j=1}^{N} b_{j,T} + b_{N+1,t}$ refers to global counterpart of $b_{j,t}$. We denote

$$H_\rho(N+1, w) + \max_{1 \leq j \leq N+1} \beta_j \frac{1}{T} \lambda_{\min}^{1/2}(V_{j,T}).$$

The above lemma says that the error $H_\rho(N, w)$ goes to zero as $N$ grows to infinity. Therefore the estimation error $\epsilon_{N,t}(\rho)$ is dominated by the “variance” term $\max_{1 \leq j \leq N+1} \beta_j \frac{1}{T} \lambda_{\min}^{1/2}(V_{j,T})$, associated with the worst past task. By relying on linear regression results (Lai & Wei, 1982) we have that $\lambda_{\min}(V_j) \geq T$. Moreover, as $\lambda_{\min}(V_j) \geq \lambda + \lambda_{\min}(V_j)$, we observe an increasing sensitivity of the incurred variance to the $\lambda$ parameter for small value of $T$. Finally, according to our choice of $\lambda = 1/T \text{Var}_w$, the suffered variance increases with the variance of our estimator. The latter in turns increases with the variance of the distribution $\rho$, which corresponds to the case in which Assumption 1 tends to be violated.
with $|A| = \sup\{|Ax| : x \in \mathbb{R}^d, \|x\| = 1\}$ the norm of matrix $A$ induced by the norm $\|\cdot\|$, which if no specified is the Euclidean norm. Finally, we denote with $\sigma_{\max}(A)$ the biggest singular value associated with matrix $A$.

### 5.1. Global Ridge Regression

In order to reduce the variance, our second approach estimates, at each round $t$ of the new sampled task $N + 1$, the mean task $\bar{w}$ as a *global ridge regression* computed over all the available samples as

$$\hat{h}_{N,t}^\lambda = \left(\tilde{V}_{N,t-1}^\lambda\right)^{-1} \tilde{b}_{N,t-1}. \quad (9)$$

Our next result provides a bound on the transfer regret of this proposed strategy.

**Theorem 3.** (Transfer Regret Bound). Let the assumptions of Lemma 2 hold and let $\hat{h}_{N,t}^\lambda$ be defined as in Equation (9). Then, the following upper bound holds

$$\mathcal{R}(T, \rho) \leq dC T \log \left(1 + \frac{T^2 L \left(\text{Var} \bar{w} + \epsilon_{N,t}(\rho)\right)}{d} \right)$$

where the mean estimation error can be bound as

$$\sqrt{\epsilon_{N,T}(\rho)} \leq \frac{S}{\lambda + \nu_{\min}} + 2(N+1) \max_{1 \leq j \leq N+1} \tilde{H}(N+1, w_j) + R \sqrt{\frac{2}{\lambda + \nu_{\min}} \log \left(T \left(1 + \frac{N T L^2}{\lambda d}\right)\right)} + H_{\rho}(N+1, \bar{w})$$

and defined $\nu_{\min} = \lambda_{\min} (\tilde{V}_{N,T})$ and we introduced

$$\tilde{H}(N, w_j) = H_{\rho}(j, w_j) \sigma_{\max}(\tilde{V}_{j,T} \tilde{V}_{N,T}^{-1})$$

which is a weighted form of the estimation error $H_{\rho}(j, w_j)$ towards the current task vector $w_j$, where the weights are defined in terms of tasks misalignment $\sigma_{\max}(\tilde{V}_{j,T} \tilde{V}_{N,T}^{-1})$.

The proof is presented in Section D of the appendix. The previous variance term $\frac{\sigma_{\max}(1/T)}{\lambda_{\min}(V_{j,T})}$ has been now replaced by $\frac{\sigma_{\max}(1/T)}{\lambda_{\min}(V_{j,T})}$. It should be easy to observe that $\nu_{\min} \leq \frac{N}{\lambda} \lambda_{\min}(V_j) \forall j \in [N]$ which leads a reduction of factor $d/N$ to the variance, which goes to zero as $N$ goes to infinity. This gain does not come for free, in fact this approach introduces a potentially high bias: $2(N+1) \max_{1 \leq j \leq N+1} \tilde{H}(N+1, w_j)$ which increases with the tasks misalignment $\sigma_{\max}(\tilde{V}_{j,T} \tilde{V}_{N,T}^{-1})$.

### 5.2. Tasks Misalignment

We now analyze the tasks misalignment factors appearing in Theorem 3, namely, the quantities $\sigma_{\max}(\tilde{V}_{j,T} \tilde{V}_{N,T}^{-1})$ and $\tilde{H}(N, w_j)$. For this purpose, we consider two opposite environments of tasks.

In the first case we assume that all the tasks parameters are equal to each other and far from the zero $d$-dimensional vector. This scenario, which corresponds to put all the mass of the task-distribution $\rho$ on a single task parameter $\bar{w}$, is clearly in agreement with Assumption 1. We expect this to be the most favorable scenario, since after completing a task, we face exactly the same task again and again. In this case, independently on the covariance matrices, whose construction also depends on the decision sets available in the different tasks, it is simple to observe that we are not suffering any bias, that is, $\tilde{H}(N, w_j) = 0$ for every $j \in [N]$ as all the task parameters are equal to each other.

The second environment is characterized by a task distribution $\rho$ that is uniform on finitely many orthogonal tasks. For instance, this is the scenario when $\rho$ is uniform distributed over the standard basis vectors $\{(S, 0, \ldots, 0), \ldots, (0, \ldots, 0, S)\} \in \mathbb{R}^d$. Differently from the previous scenario, here after completing a task we will probably face an orthogonal task. It should be quite natural to see that this is the most unfavorable case and to expect to not have transfer learning between tasks. This is confirmed by the regret bound due to the misalignment expressed by the covariance matrices $\sigma_{\max}(\tilde{V}_{j,T} \tilde{V}_{N,T}^{-1})$. Indeed, since we can have at most $d$ misaligned arms, we have the following upper bound $\tilde{H}$ to the term $\sigma_{\max}(\tilde{V}_{j,T} \tilde{V}_{N,T}^{-1})$. Based on these observations we can conclude that the bigger the cardinality of the set of basis induced by the distribution $\rho$, the larger the number of completed tasks required to have a proper transfer. We will now focus on an intermediate case satisfying Assumption 1. In order to control the term $\sigma_{\max}(\tilde{V}_{j,T} \tilde{V}_{N,T}^{-1})$ and to give the possibility to generate aligned matrices when dealing with similar tasks, we introduce an additional mild assumption:

**Assumption 2.** (Shared Induced Basis) The decision sets are shared among all the tasks and tasks sampled according to Assumption 1 induces that the covariance matrices generated by running the BIAS-OFUL algorithm (Algorithm 1) share the same basis:

$$V_i = P \Sigma_i P^*, \quad \forall i \in [N]. \quad (10)$$

This assumption is quite mild as it just states that similar tasks share the same pulled arms with no restrictions on the pulling frequency. This is the case when the decision set is fixed among different rounds and tasks, that is, $D_{j,t} = D \forall j \in [N]$ and $t \in [T]$, and consists of $d$ orthogonal arms. If Assumption 2 is satisfied, then we can obtain the following bound: $\sigma_{\max}(\tilde{V}_{j,T} \tilde{V}_{N,T}^{-1}) \leq 1$. Furthermore, if we denote by $M$ the number of tasks necessary to achieve a stationary behavior of the BIAS-OFUL policy in terms of covariance matrices, then $\sigma_{\max}(\tilde{V}_{j,T} \tilde{V}_{N,T}^{-1}) \leq 1/(N - M)$. 

Finite-dimension linear bandits
5.3. Smallest Global Eigenvalue $\nu_{\min}$

It only remains to analyze the term $\nu_{\min}$. We observe that it satisfies the lower bound

$$\nu_{\min} = \lambda_{\min} \left( \sum_{j=1}^{N+1} V_{j,T} \right) \geq \sum_{j=1}^{N+1} \lambda_{\min} \left( V_{j,T} \right)$$

$$\geq (N + 1) \log T$$

where in the last step we have relied on linear regression result from (Lai & Wei, 1982) which shows that the condition $O(\lambda_{\min}) = \log(\lambda_{\max})$ is required to guarantee asymptotic consistency, necessary to have sublinear anytime regret. Since $\min_{j \in [N]} \lambda_{\max}(V_j) = O(T)$, this condition implies that $\min_{j \in [N]} \lambda_{\min}(V_j) \geq \log T$. 

6. Experiments

In this section we test the real effectiveness of the proposed approaches. The theoretical results stated that the method presented in Section 4 does not introduce any bias but it may incur an additional variance according to the variance of the task-distribution $\text{Var}_\rho$. On the contrary, the solution proposed in Section 5 which massively uses all the observed samples together, reduces the variance (at least) by a factor $d/N$, at the price of an extra bias term.

As it was mentioned in Section 3, the parameter $w^*$ associated to each single task is unknown, therefore we cannot compute the gap $\|\hat{h}^k - w^*\|_2$ defining the term $\beta^k_1(1/T)$. The main issue is that according to Equation 4, in order to pick the next arm, it seems that the algorithm needs to compute its exact value. However, we can simply split the norm and rely on the assumption that $\|w^*\| \leq S$, so to remove the dependency on $w^*$. Indeed, it is important to emphasize that the real knowledge transfer happens in terms of $w^\circ$, see Equation 5. This can be noticed by observing that the gap $\|\hat{h}^k - w^*\|$ equally affects all the available arms.

6.1. Experimental Results

In all the presented experiments the policy OPT knows the parameter $w_j$ associated to task $j$ and picks the next arm as $x_{j,t} = \arg\max x \in d_j \sum \xi^T w_j$. The policies AVG-OFUL and RR-OFUL implement Algorithms 1 and 2 and estimate $\hat{h}$ as per Equations 8 and Equation 9, respectively. The Oracle policy knows the mean task parameter $w$ and uses it as the bias $h$ in BIAS-OFUL (Corollary 2 (ii)). Analogously, the ITL policy consists of BIAS-OFUL with bias set equal to 0, see Corollary 2 (i). The regularization hyper-parameter $\lambda$ was selected over a logarithmic scale. We will start by considering a pair of synthetic experiments in which we show how the hyper-parameter $\lambda$ affects the performance. We then present experiments on two real datasets. We will denote with $K$ the size of the decision set $D$.

**Figure 1.** Cumulative reward measured after $N = 10$ tasks and averaged over 10 independent test tasks, with $\lambda = 1$.

**Figure 2.** Cumulative reward measured after $N = 10$ tasks and averaged over 10 independent test tasks, with $\lambda = 100$.

**Synthetic Data** Similarly to what was done in (Denevi et al., 2019), we first generated an environment of tasks in which running the Oracle policy is expected to outperform the ITL approach. In agreement with Assumption 1, we sample the task vectors from a distribution characterized by a much smaller variance than its second moment. That is, each task parameter $w_j$ is sampled from a Gaussian distribution with mean $w$ given by the vector in $\mathbb{R}^d$ with all components equal to 1 and $\text{Var}_\rho = 1$. As far as the decision set concerns, we first generate a random square matrix $P$ with size $d$ and then compute its qr factorization $P = QR$, where $Q$ is a matrix with orthonormal columns and $R$ is an upper-triangular matrix. We then associate to each base arm the direction associated to a column of the matrix $Q$. This will guarantee having arms that are almost orthogonal each other. Finally, at each round $t \in [T]$ the decision set $D_t$ is initialized as a set of $K$ random vector that are first shifted towards the respective arm base direction and then normalized. Notice that by following this generation mechanism we avoid any inductive bias between the task vectors and the arms ones, as they are actually
independent. Each task consists of $T = 50$ rounds, in which we have $K = 5$ arms of size $d = 20$. In order to generate the rewards, we first compute the inner product between the user (task) vector and the arm (input) vector, we shift the resulting output interval $[0, 1]$ and then add to a Gaussian noise $\mathcal{N}(0, 1)$, to compute the rewards. Finally, we assigned reward $1$ to the arm having the maximum final reward, $0$ to the others. In Figures 1 and 2, we report the results generated with $\lambda = 1$ and $\lambda = 100$, respectively. It is easy to observe that the stronger the regularization, the more the AVG-OFUL tends to the Oracle. Conversely, RR-OFUL get penalized with the increasing of $\lambda$, due to its bias.

**LastFM Data** The first dataset we considered is extracted from the music streaming service Last.fm (Cantador et al., 2011). It contains 1892 possible users and 17632 artists. This dataset contains information about the artists listened by a given user, and we used this information to define the payoff function. We first removed from the set of items those with less than 30 ratings and then we repeat the same procedure for the users. This operation yields an user rating matrix of size $741 \times 538$. Starting from this reduced matrix we derived the arms and the users vectors by computing an SVD decomposition where we kept only the first $30$ features associated to the users and to the items. In order to consider tasks satisfying Assumption 1, we randomly pick an user and compute the set of its $N = 20$ most similar users according to the l2-distance between their vectors. Each task lasts $T = 5$ rounds and consists of $K = 5$ arms. At each round $t$, the decision set consists of one arm whose rating was at least equal to $4$ and $K - 1$ arms whose ratings were at most equal to $3$. The rewards were then generated analogously to the synthetic case. The Oracle policy knows $\mathbf{w}$ which is computed as the average between the $\mathcal{N} = 20$ most similar users according to the l2-distance between their vectors. In Figure 3 (and Figure 4) we displayed the cumulative regret suffered with respect to the optimal policy, which during each task $j \in [N]$ knows the true user parameter $\mathbf{w}_j$. The vertical yellow lines indicate the end of each task. From the presented results we can observe that both the proposed policies AVG-OFUL and RR-OFUL outperform the ITL approach, while the Oracle policy is consistent with Corollary 2 and Assumption 1.

**Movielens** Here we consider the Movielens data (Harper & Konstan, 2015). It contains 1M anonymous ratings of approximately 3900 movies made by 6040 users. As before we first removed from the set of movies those with less than 500 ratings, and from the set of users those with less than 200 rated movies. This preprocessing procedure yields an user rating matrix of size $847 \times 618$. Unlike the Last.fm case, here adopting SVD to generate the arm/user vectors seems not appropriate. Indeed, by exploring the retrieved singular values, we could not find a subspace which provides a good approximation of the real ratings unless we keep all the latent features. Therefore, in order to find a set of similar users we observe better results by using the KMeans clustering algorithm over the user vectors. The results displayed in Figure 4 were generated by running KMeans with $C = 20$ clusters with user vectors of size $d = 10$. We then picked all the resulting clusters by filtering out the clusterings with a silhouette value lower than 0.15 and for each cluster of the clustering we have discarded those with less than 20 users. Furthermore, in order to let the tasks be simpler, we reduced the variance of the noisy components affecting rewards to 0.1. The difficulty in finding a valid set of similar tasks yields a high task misalignment, which is confirmed by the fact that the best performance occur for small value of $\lambda$. Indeed, Figure 4 considers $\lambda = 1$. Here the AVG-OFUL policy behaves almost equally to the ITL approach, conversely, the task misalignment caused bad performances to the RR-OFUL policy, confirming its higher sensitivity to task dissimilarity (see Theorem 3).

7. Conclusions and Future Work

In this work we studied a meta-learning framework with stochastic linear bandit tasks. We have first introduced a novel regularized version of OFUL, where the regularization depends on the Euclidean distance to a bias vector. We showed that setting appropriately the bias leads a substantial improvement compared to learning each task in isolation. This observation motivated two alternative approaches to estimate this bias: while the first one may suffer a potentially
high variance, the second might incur a strong bias.

In the future, it would be valuable to investigate the existence of unbiased estimators which do not suffer any variance. Furthermore, while in our analysis we set $\lambda = 1/T\text{Var} h$, in the future it would be also interesting to learn its value as part of the learning problem. Experimentally, we observed that when Assumption 1 is satisfied, adopting the unbiased estimator yields better results than the second one, which is biased. One more direction of future research would be to extend other meta-learning approaches, such as those based on feature sharing, to the banding setting. Finally, a problem which remains to be studied is the combination of meta-learning with non-stochastic bandits.

References


Azar, M. G., Lazaric, A., and Brunskill, E. Sequential transfer in multi-armed bandit with finite set of mod-


Supplemental Material

A. Proof of Theorem 1

Proof. Starting from the biased-regularized estimation of Equation 5,
\[
\hat{w}_t^h = (V_t^\lambda) \cdot^{-1} X_t (Y_t - X_t h) + h = (V_t^\lambda) \cdot^{-1} X_t (X_t w^* + \eta_t - X_t h) + h
\]
\[
= (V_t^\lambda) \cdot^{-1} X_t X_t w^* + (V_t^\lambda) \cdot^{-1} X_t \eta_t - (V_t^\lambda) \cdot^{-1} X_t X_t h + h
\]
Given this construction we can obtain the following equalities:
\[
\hat{w}_t^h - w^* = (V_t^\lambda) \cdot^{-1} X_t \eta_t + h - (V_t^\lambda) \cdot^{-1} X_t X_t h - \lambda (V_t^\lambda) \cdot^{-1} w^*
\]
\[
= (V_t^\lambda) \cdot^{-1} X_t \eta_t + (\lambda (V_t^\lambda) \cdot^{-1}) (h - w^*)
\]
Then, for any \( x \in \mathbb{R}^d \) the following holds:
\[
x^\top (\hat{w}_t^h - w^*) = \langle x, X_t \eta_t \rangle (V_t^\lambda) \cdot^{-1} + \lambda \langle x, h \rangle (V_t^\lambda) \cdot^{-1} - \lambda \langle x, w^* \rangle (V_t^\lambda) \cdot^{-1}
\]
\[
\leq \|x\| (V_t^\lambda) \cdot^{-1} (\|X_t \eta_t\| (V_t^\lambda) \cdot^{-1} + \lambda \|h - w^*\| (V_t^\lambda) \cdot^{-1})
\]
where in the last step we have applied Cauchy-Schwarz inequality. Plugging in \( x = V_t^\lambda (\hat{w}_t^h - w^*) \) we obtain:
\[
\|\hat{w}_t^h - w^*\|_2^2 \leq \|\hat{w}_t^h - w^*\|_\lambda (\|X_t \eta_t\|_\lambda (\|V_t^\lambda\|^{-1}) + \lambda \|h - w^*\|_\lambda^{-1})
\]
Finally we divide both sides by \( \|\hat{w}_t^h - w^*\|_\lambda \) we obtain:
\[
\|\hat{w}_t^h - w^*\|_\lambda \leq \|X_t \eta_t\|_\lambda (\|V_t^\lambda\|^{-1}) + \lambda \|h - w^*\|_\lambda^{-1}.
\]
Finally we bound the noisy term \( \|X_t \eta_t\|_\lambda^{-1} \) by leveraging on Theorem 1 of (Abbasi-Yadkori et al., 2011), obtaining:
\[
\|\hat{w}_t^h - w^*\|_\lambda \leq R \left( 2 \log \left( \frac{\det (V_t^\lambda)^{1/2}}{\det(\lambda T)^{1/2} \delta} \right) + \lambda^{1/2} \|h - w^*\|_2 \right)
\]
(11)
where we have used the fact that: \( \|h - w^*\|_\lambda^{-1} \leq \frac{1}{\lambda_{\min}(V_t^\lambda)} \|h - w^*\|_2 \leq \frac{1}{\lambda} \|h - w^*\|_2 \).

B. Proof of Lemma 1

Proof. We start by analysing the instantaneous regret as follows:
\[
r_t = \langle w^*, x_t \rangle - \langle w^*, x_t^h \rangle = \langle w^*, x_t \rangle - \langle \hat{w}_t^h, x_t^h \rangle + \langle \hat{w}_t^h, x_t^h \rangle - \langle w^*, x_t^h \rangle
\]
\[
\leq \langle \hat{w}_t^h, x_t \rangle - \langle w^*, x_t^h \rangle = \langle \hat{w}_{t-1}^h - w^*, x_t^h \rangle + \langle \hat{w}_t^h - \hat{w}_{t-1}^h, x_t^h \rangle
\]
\[
\leq \|\hat{w}_{t-1}^h - w^*\|_\lambda^{-1} \|x_t^h\|_\lambda^{-1} + \|\hat{w}_t^h - \hat{w}_{t-1}^h\|_\lambda^{-1} \|x_t^h\|_\lambda^{-1} \leq 2 \beta_t^h(\delta) \|x_t^h\|_\lambda^{-1}
\]
where in the first inequality we have leveraged on the fact that \( \hat{w}_{t-1}^h \) is optimistic and in the last the ellipsoid bound specified in Equation 11. The bound of the cumulative regret follows from the bound of (Abbasi-Yadkori et al., 2011), hence with probability at least \( 1 - \delta \), for all \( T \geq 0 \):
\[
R(T, w^*) \leq \sqrt{T \sum_{t=1}^{T} r_t^2} \leq 4 \sqrt{T \log \left( \frac{\det (V_t^\lambda)}{\det(\lambda T)^{1/2} \delta} \right)} - \log \left( \frac{\det(\lambda I)}{\beta_t^h(\delta)} \right)
\]
\[
\leq 4 \sqrt{T d \log \left( 1 + \frac{TL}{\lambda d} \right)} \left( \lambda^{1/2} \|w^* - h\|_2 + R \sqrt{2 \log(1/\delta) + d \log (1 + TL/(\lambda d))} \right)
\]
where the last two steps follow from Lemma 11 of (Abbasi-Yadkori et al., 2011) and the definition of \( \beta_t^h(\delta) \) (Equation 11). The stated result is derived analogously to Corollary 19.3 of (Lattimore & Szepesvári, 2020) considering \( \delta = 1/T \).
C. Proof of Corollary 1

Proof. We start by considering the oracle scenario which is given by $h = w^*$.

$$
\lim_{\lambda \to \infty} \left[ C \sqrt{T d \log \left( 1 + \frac{TL}{\lambda d} \right)} \left( R \sqrt{d \log(T + T^2 L/(\lambda d))} \right) \right] = C \sqrt{T d \log(1)} \left( R \sqrt{d \log(T + T^2 L/(\lambda d))} \right) = 0
$$

As far as the independent task learning scenario concerns, the following holds:

$$
\lim_{\lambda \to \infty} C \sqrt{T d \log \left( 1 + \frac{TL}{\lambda d} \right)} \left( \lambda S + R \sqrt{d \log(T + T^2 L/(\lambda d))} \right) = \lim_{\epsilon \to 0} C \sqrt{T d \log \left( 1 + \epsilon \right)} \left( S \sqrt{\frac{TL}{\epsilon d}} + R \sqrt{d \log(T + T^2 L/(\lambda d))} \right)
$$

where we have used the substitution $\epsilon = \frac{TL}{\lambda d}$ and the fact that $\lim_{\epsilon \to 0} \frac{\log(1+\epsilon)}{\epsilon} \to 1$. \qed

D. Proof of Theorem 3

We start by presenting two Lemmas which are necessary to obtain the final bound. Firstly, we need to introduce an additional variable:

$$
\tilde{h}_{N,t+1} = (\tilde{V}_{N,t})^{-1} \left( \sum_{j=1}^{N} V_{j,T}w_j + V_{N+1,t}w_{N+1} \right)
$$

We will then split the analysis by studying separately the estimation error $\tilde{h}_{N,t+1} - \tilde{h}_{N,t+1}$ (Lemma 4) and the estimation bias $\tilde{h}_{N,t+1} - \tilde{h}_{N,t+1}$ (Lemma 5).

Lemma 4. The following rewriting holds:

$$
\tilde{h}_{N,t+1} - \tilde{h}_{N,t+1} = (\tilde{V}_{N,t})^{-1} \left( \sum_{j=1}^{N} \sum_{s=1}^{T} x_{j,s} \eta_{j,s} + \sum_{s=1}^{t} x_{N+1,s} \eta_{N+1,s} \right) - \lambda (\tilde{V}_{N,t})^{-1} \tilde{h}_{N,t+1}
$$

Proof.

$$
\tilde{h}_{N,t+1} = (\tilde{V}_{N,t})^{-1} \tilde{b}_{N,t} = (\tilde{V}_{N,t})^{-1} \left( \sum_{j=1}^{N} \sum_{s=1}^{T} x_{j,s} \eta_{j,s} + \sum_{s=1}^{t} x_{N+1,s} \eta_{N+1,s} \right)
$$

$$
= (\tilde{V}_{N,t})^{-1} \left( \sum_{j=1}^{N} \sum_{s=1}^{T} x_{j,s} \left( x_{j,s}^T w_j + \eta_{j,s} \right) + \sum_{s=1}^{t} x_{N+1,s} \eta_{N+1,s} \right)
$$

$$
= (\tilde{V}_{N,t})^{-1} \left( \sum_{j=1}^{N} \sum_{s=1}^{T} x_{j,s} \eta_{j,s} + \sum_{s=1}^{T} x_{N+1,s} \eta_{N+1,s} \right) + (\tilde{V}_{N,t})^{-1} \left( \sum_{j=1}^{N} \sum_{s=1}^{T} x_{j,s} x_{j,s}^T w_j + \sum_{s=1}^{t} x_{s} x_{N+1,s}^T w_{N+1} \right)
$$

$$
= (\tilde{V}_{N,t})^{-1} \left( \sum_{j=1}^{N} \sum_{s=1}^{T} x_{j,s} \eta_{j,s} + \sum_{s=1}^{t} x_{N+1,s} \eta_{N+1,s} \right) +
$$
Meta-learning with Stochastic Linear Bandits

\[
\begin{align*}
&\left(\tilde{V}_{N,t}^\lambda\right)^{-1} \tilde{V}_{N,t} \left(\tilde{V}_{N,t}\right)^{-1} \left(\sum_{j=1}^{N} \sum_{s=1}^{T} x_{j,s} x_{j,s}^T w_j + \sum_{s=1}^{t} x_{N+1,s} x_{N+1,s}^T w_{N+1}\right) \\
&= \left(\tilde{V}_{N,t}^\lambda\right)^{-1} \left(\sum_{j=1}^{N} \sum_{s=1}^{T} x_{j,s} \eta_{j,s} + \sum_{s=1}^{t} x_{N+1,s} \eta_{N+1,s}\right) + \left(\tilde{V}_{N,t}^\lambda\right)^{-1} \tilde{V}_{N,t} \left(\tilde{V}_{N,t}\right)^{-1} \left(\sum_{j=1}^{N} v_j^T w_j + V_{N+1,t} w_{N+1}\right) \\
&= \left(\tilde{V}_{N,t}^\lambda\right)^{-1} \left(\sum_{j=1}^{N} \sum_{s=1}^{T} x_{j,s} \eta_{j,s} + \sum_{s=1}^{t} x_{N+1,s} \eta_{N+1,s}\right) + \tilde{V}_{N,t} \tilde{\eta}_{N,t+1} + \lambda \left(\tilde{V}_{N,t}^\lambda\right)^{-1} \left(\tilde{h}_{N,t+1} - \tilde{h}_{N,t+1}\right)
\end{align*}
\]

which gives the claimed result.

**Lemma 5.** According to what we have done in Section 4, we use:

\[
\tilde{h}_{N,t+1} = \frac{1}{NT + t} \left(\sum_{j=1}^{N} T w_j + t w_{N+1}\right).
\]

Differently from \(\tilde{h}_{N,t}\) this definition is a weighted average of the vectors of the \(N\) completed tasks. Hence, we have:

\[
\|w - \tilde{h}_{N,t}\| \leq \frac{1}{NT + t} \sum_{j=1}^{N} \left(\|w - \tilde{h}_{N,t}\| + (NT + t) \|\tilde{h}_{N,t} - \tilde{h}_{N,t}\|\right)
\]

\[
= H_\rho(N + 1, w) + \|\tilde{h}_{N,t} - \tilde{h}_{N,t}'\|
\]

where we have denoted with \(H_\rho(N + 1, w)\) a according to what we have done in Section 4. We can now focus on the term \(\|\tilde{h}_{N,t} - \tilde{h}_{N,t}'\|\) which can be equivalently rewritten as follows:

\[
\|\tilde{h}_{N,t+1} - \tilde{h}_{N,t+1}\| = \left\|\left(\tilde{V}_{N,t}\right)^{-1} \sum_{j=1}^{N} \left(V_{j,T} w_j + V_{N+1,t} w_{N+1}\right) - \tilde{h}_{N,t}\right\|
\]

\[
\leq \sum_{j=1}^{N} \tilde{V}_{N,t}^{-1} V_{j,T} \|w_j - \tilde{h}_{N,t}\| + \tilde{V}_{N,t}^{-1} \|V_{N+1,t} w_{N+1} - \tilde{h}_{N,t}\|
\]

\[
\leq \sum_{j=1}^{N} H_\rho(N + 1, w_j) \tilde{V}_{N,t}^{-1} V_{j,T} + H_\rho(N + 1, w) \tilde{V}_{N,t}^{-1} V_{j,T}
\]

\[
= \sum_{j=1}^{N} H_\rho(N + 1, w_j) \sigma_{\max} \left(V_{j,T} \tilde{V}_{N,t}^{-1}\right) + H_\rho(N + 1, w_{N+1}) \sigma_{\max} \left(V_{j,T} \tilde{V}_{N,t}^{-1}\right)
\]

\[
\leq (N + 1) \max_{j=1,\ldots,N+1} \left(H_\rho(N + 1, w_j) \sigma_{\max} \left(V_{j,T} \tilde{V}_{N,t}^{-1}\right)\right) = (N + 1) \max_{j=1,\ldots,N+1} \tilde{H}(N + 1, w_j)
\]

We have used the fact that the matrix norm of a given matrix \(A\) induced by the Euclidean norm corresponds to the spectral norm, which is the largest singular value of the matrix \(\sigma_{\max}(A)\).

**D.1. Proof of Theorem 3**

We start the analysis from the result of Lemma 2:

\[
R(T, \rho) \leq d \sqrt{\frac{T^2 L \left(\mathbb{E}_{w \sim \rho} \left[\|w - h\|_2^2\right]\right)}{1 + \frac{T \log \left(\frac{T^2 L \left(\mathbb{E}_{w \sim \rho} \left[\|w - h\|_2^2\right]\right)}{d}\right)}}}
\]
we can then set the hyperparameter $h = \hat{H}_{N,T}$ and focusing on the first term in brackets we obtain:

\[
\sqrt{\mathbb{E}_{\mathbf{w} \sim \rho} \left[ \left\| \mathbf{w} - \hat{h}_{N,T}^\lambda \right\|^2 \right]} \leq \sqrt{\text{Var} \mathbf{w}} + \sqrt{\epsilon_{N,t}(\rho)}
\]

According to Lemma 5 the following rewriting holds:

\[
\sqrt{\epsilon_{N,t}(\rho)} \leq H_{\rho}(N + 1, \overline{\mathbf{w}}) + (N + 1) \max_{j=1, \ldots, N+1} \tilde{H}(N+1, j) + \left\| \hat{h}_{N,T} - \hat{h}_{N,T}^\lambda \right\|_2
\]

It remains only to apply Lemma 4 which gives:

\[
\left\| \hat{h}_{N,T} - \hat{h}_{N,T}^\lambda \right\|_2 = \left\| \left( \hat{V}_{N,T}^\lambda \right)^{-1} \left( \sum_{j=1}^{N} \sum_{s=1}^{T} x_{j,s} \eta_{j,s} + \sum_{s=1}^{T} x_{s} \eta_{s} \right) \right\|_2 + \left\| \lambda \left( \hat{V}_{N,T}^\lambda \right)^{-1} \hat{h}_{N,T} \right\|_2
\]

\[
\leq \sum_{j=1}^{N} \sum_{s=1}^{T} x_{j,s} \eta_{j,s} + \sum_{s=1}^{T} x_{s} \eta_{s} \right\|_2 \left( \hat{V}_{N,T}^\lambda \right)^{-2} + \lambda \left\| \hat{h}_{N,T} \right\|_2 \left( \hat{V}_{N,T}^\lambda \right)^{-2}
\]

\[
\leq \frac{1}{\lambda_{\min}^2(\hat{V}_{N,T}^\lambda)} \left\| \sum_{j=1}^{N} \sum_{s=1}^{T} x_{j,s} \eta_{j,s} + \sum_{s=1}^{T} x_{s} \eta_{s} \right\|_2 + \frac{1}{\lambda_{\min}(\hat{V}_{N,T}^\lambda)} \left\| \hat{h}_{N,T} \right\|_2
\]

\[
\leq \frac{1}{\lambda_{\min}^2(\hat{V}_{N,T}^\lambda)} R \sqrt{2 \log \left( T + \frac{(N+T)TL^2}{\lambda d} \right)} + \left\| \hat{h}_{N,T} - \hat{h}_{N,T}^\lambda \right\|_2 + \frac{1}{\lambda_{\min}(\hat{V}_{N,T}^\lambda)} \left\| \hat{h}_{N,T} \right\|_2
\]

\[
\leq \frac{1}{\lambda_{\min}^2(\hat{V}_{N,T}^\lambda)} R \sqrt{2 \log \left( T + \frac{(N+T)TL^2}{\lambda d} \right)} + (N + 1) \max_{j=1, \ldots, N+1} \tilde{H}(N+1, j) + \frac{S}{\lambda_{\min}(\hat{V}_{N,T}^\lambda)}
\]

where in the last inequality we have applied once more Lemma 5. We can now introduce $\nu_{\min} = \lambda_{\min}(\hat{V}_{N,T}^\lambda)$ as the minimum eigenvalue of the global covariance matrix without regularization which gives the following bound:

\[
\frac{1}{\lambda_{\min}(\hat{V}_{N,T}^\lambda)} \leq \frac{1}{\lambda + \nu_{\min}}
\]

Putting everything together gives the claimed result:

\[
\sqrt{\epsilon_{N,T}(\rho)} \leq H_{\rho}(N+1, \overline{\mathbf{w}}) + 2(N+1) \max_{j=1, \ldots, N+1} \tilde{H}(N+1, \mathbf{w}_j) + \frac{1}{(\lambda + \nu_{\min})^2} R \sqrt{2 \log \left( T + \frac{(N+T)TL^2}{\lambda d} \right)} + \frac{S}{\lambda + \nu_{\min}}
\]