Multi-Task Off-Policy Learning from Bandit Feedback

Joey Hong [1] Branislav Kveton [2] Manzil Zaheer [3] Sumeet Katariya [2] Mohammad Ghavamzadeh [4]

Abstract

Many practical problems involve solving similar tasks. In recommender systems, the tasks can be users with similar preferences; in search engines, the tasks can be items with similar affinities. To learn statistically efficiently, the tasks can be organized in a hierarchy, where the task affinity is captured using an unknown latent parameter. We study the problem of off-policy learning for similar tasks from logged bandit feedback. To solve the problem, we propose a hierarchical off-policy optimization algorithm `HierOPO`. The key idea is to estimate the task parameters using the hierarchy and then act pessimistically with respect to them. To analyze the algorithm, we develop novel Bayesian error bounds. Our bounds are the first in off-policy learning that improve with a more informative prior and capture statistical gains due to hierarchical models. Therefore, they are of a general interest. `HierOPO` also performs well in practice. Our experiments demonstrate the benefits of using the hierarchy over solving each task independently.

1. Introduction

Many interactive systems, such as search and recommender systems, can be modeled as a *contextual bandit* (Li et al., 2010; Chu et al., 2011), where a *policy* observes a *context*, takes one of possible *actions*, and then receives a *stochastic reward* for that action. In many applications, it is prohibitively expensive to learn policies online by contextual bandit algorithms, because exploration has a major impact on user experience. However, offline data collected by a previously deployed policy are often available. Offline, or *off-policy*, optimization using such logged data is a practical way of learning policies without costly online interactions (Dudik et al., 2014; Swaminathan & Joachims, 2015). Because we cannot explore beyond the logged dataset, it is important to use the logged data in the most statistically efficient way. One way of achieving this is by modeling the structure of the solved problem. As an example, in bandit algorithms, we could achieve higher statistical efficiency by using information about the reward distribution (Garivier & Cappe, 2011), a prior distribution over model parameters (Thompson, 1933; Agrawal & Goyal, 2012; Chapelle & Li, 2012; Russo et al., 2018), or features (Dani et al., 2008; Abbasi-Yadkori et al., 2011; Agrawal & Goyal, 2013). In this work, we consider a natural structure where multiple similar tasks are related through a *hierarchical Bayesian model* (Gelman et al., 2013; Kveton et al., 2021; Hong et al., 2022b). Each task is parameterized by a *task parameter* sampled i.i.d. from a distribution parameterized by a *hyper-parameter*. The unknown hyper-parameter relates the tasks. Specifically, data from any task helps in improving its estimate, which in turn improves estimates of all other task parameters.

Although the tasks are similar, they are sufficiently different to require different polices, and we address this multi-task off-policy learning problem in this work. To solve the problem, we propose an algorithm called hierarchical off-policy optimization (`HierOPO`). Since off-policy algorithms must reason about counterfactual rewards of actions that may not have been taken frequently in the logged dataset, a common approach is to learn pessimistic, or *lower confidence bound (LCB)*, estimates of the mean rewards and act according to them (Buckman et al., 2021; Jin et al., 2021). `HierOPO` is an instance of this approach where high-probability LCBs are estimated using a hierarchical model.

Our work makes four major contributions. First, we formalize the problem of multi-task off-policy optimization as a multi-task bandit in a hierarchical model. Second, we propose an efficient algorithm for solving the problem, which we call `HierOPO`. The key idea in `HierOPO` is to compute lower confidence bounds on the mean rewards of actions and act according to them. The LCBs can be computed in a closed form in linear Gaussian models (Lindley & Smith, 1972). Third, we analyze the quality of our policies using Bayesian error bounds. Our bounds capture the effect of a more informative prior and statistical gains due to hierarchical models. These are the first such bounds in off-policy learning and should be of a wide interest. Finally, we evalu-

[1] University of California, Berkeley [2] Amazon [3] DeepMind [4] Google Research. Correspondence to: Joey Hong <joey_hong@berkeley.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

ate HierOPO on both synthetic and real-world problems.

2. Setting

We start with introducing our notation. Random variables are capitalized, except for Greek letters like θ. For any positive integer n, we define $[n] = \{1, \dots, n\}$. The indicator function is $\mathbb{1}\{\cdot\}$. The i-th entry of vector v is denoted by v_i. If the vector is already indexed, such as v_j, we write $v_{j,i}$. We denote the maximum and minimum eigenvalues of matrix $M \in \mathbb{R}^{d \times d}$ by $\lambda_1(M)$ and $\lambda_d(M)$, respectively.

In the classic contextual bandit (Li et al., 2010), the agent observes a *context* $x \in \mathcal{X}$, where $\mathcal{X}$ is a *context set*; takes an *action* $a \in \mathcal{A}$, where $\mathcal{A}$ is an *action set*; and observes a *stochastic reward* $Y \sim P(\cdot \mid x, a; \theta)$, where $P(\cdot \mid x, a; \theta)$ is the *reward distribution* parameterized by a *model parameter* $\theta \in \Theta$. We denote the mean reward of action a in context x under parameter θ by $r(x, a; \theta) = \mathbb{E}_{Y \sim P(\cdot \mid x, a; \theta)}[Y]$ and assume that the rewards are σ^2-sub-Gaussian.

2.1. Multi-Task Bandit

In this paper, we simultaneously solve m similar contextual bandit instances, which we call *tasks*. Therefore, our problem is a *multi-task contextual bandit* (Azar et al., 2013; Deshmukh et al., 2017; Cella et al., 2020; Kveton et al., 2021; Moradipari et al., 2022). The set of all tasks is denoted by $\mathcal{S}$ and we index the tasks by $s \in \mathcal{S}$. The reward distribution in task s is parameterized by a *task parameter* $\theta_{s,*} \in \Theta$, which is sampled i.i.d. from a *task prior distribution* $\theta_{s,*} \sim P(\cdot \mid \mu_*)$. The task prior is parameterized by an unknown *hyper-parameter* μ_*, which is sampled from a *hyper-prior* Q. The hyper-prior represents the agent's prior knowledge about μ_*. In a recommender system, the task could be a user, the task parameter could be their preferences, and the hyper-parameter could be the preferences of an average user. We use this example in our experiments in Section 7. A similar setup was studied in the online setting by Hong et al. (2022b).

Unlike prior works in multi-task bandits, we focus on the offline setting where we learn task-specific policies from logged data. One distinguishing characteristic of our setting is that each task has its own parameter $\theta_{s,*}$, and thus may require a different policy. As a result, we learn a separate *policy* $\pi_s : \mathcal{X} \to \mathcal{A}$ for each task s. To simplify notation, we focus on deterministic policies. Our results can be extended to stochastic policies by accounting for an additional expectation over actions. We denote the set of *stationary deterministic policies* by $\Pi = \{\pi : \mathcal{X} \to \mathcal{A}\}$.

We learn the policies from a *logged dataset* of size n. The dataset is given by $\mathcal{D} = \{(S_t, X_t, A_t, Y_t)\}_{t \in [n]}$, where S_t is a task, $X_t \sim P_{\mathsf{x}}$ is a context, $A_t \sim \pi_0(X_t)$ is an action, and $Y_t \sim P(\cdot \mid X_t, A_t; \theta_{S_t,*})$ is a reward in interaction t. Here P_{x} is a *context distribution* and $\pi_0 \in \Pi$ is a *logging policy*. To simplify notation, we assume that π_0 is the same for all tasks. It can be stochastic. A graphical model of our setting, which shows dependencies among all variables in our problem, is in Figure 1. We do not assume that π_0 is known, although this assumption is common (Dudik et al., 2014; Swaminathan & Joachims, 2015).

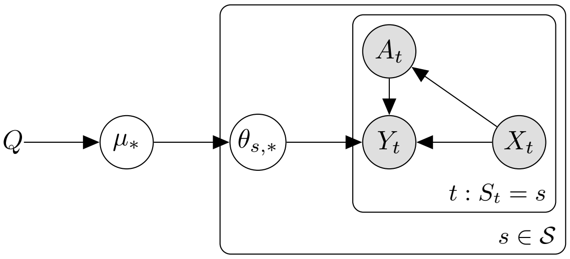

Figure 1. A graphical model of a multi-task contextual bandit.

2.2. Objective

The *value of policy* $\pi_s \in \Pi$ in task $s \in \mathcal{S}$ with parameter $\theta_{s,*}$ is defined as

$$V(\pi_s; \theta_{s,*}) = \mathbb{E}\left[r(X, \pi_s(X); \theta_{s,*})\right] ,$$

where the randomness is only over context $X \sim P_{\mathsf{x}}$. The *optimal policy* $\pi_{s,*}$ is defined as

$$\pi_{s,*} = \arg\max_{\pi \in \Pi} V(\pi; \theta_{s,*}) .$$

Let $\hat{\pi}_s \in \Pi$ be some estimated optimal policy from logged dataset $\mathcal{D}$. A standard approach in off-policy optimization is to derive an (ε, δ) bound

$$V(\hat{\pi}_s; \theta_{s,*}) \geq V(\pi_{s,*}; \theta_{s,*}) - \varepsilon , \tag{1}$$

which holds with probability at least $1 - \delta$ for a specified maximum error ε (Strehl et al., 2010; Li et al., 2018). The bound says that the policy $\hat{\pi}_s$ is at most ε worse than the optimum $\pi_{s,*}$ with a high probability. The error ε depends on δ, $\mathcal{D}$, $\hat{\pi}_s$, and problem hardness. Such bounds can be derived using concentration inequalities for sub-Gaussian rewards (Boucheron et al., 2013), under the assumptions that the parameter $\theta_{s,*}$ is fixed and bounded. We call this setting *frequentist*.

In our work, we study a *Bayesian* setting, where the prior distribution of $\theta_{s,*}$ and dataset $\mathcal{D}$ allow the agent to derive the posterior distribution of $\theta_{s,*}$, $\hat{P}_s(\theta) = \mathbb{P}\left(\theta_{s,*} = \theta \,|\, \mathcal{D}\right)$. To model that $\theta_{s,*}$ is random, and that the prior and dataset $\mathcal{D}$ provide additional information about $\theta_{s,*}$, it is natural to guarantee (1) in expectation over the posterior of $\theta_{s,*}$. We formalize this as an (ε, δ) bound

$$\mathbb{P}\left(V(\hat{\pi}_s; \theta_{s,*}) \geq V(\pi_{s,*}; \theta_{s,*}) - \varepsilon \,|\, \mathcal{D}\right) \geq 1 - \delta , \tag{2}$$

where ε is a specified maximum error that depends on δ, $\mathcal{D}$, $\hat{\pi}_s$, and problem hardness. The main difference from (1) is that $\theta_{s,*}$, and thus also $\pi_{s,*}$, are random.

The Bayesian view allows us to derive (ε, δ) error bounds with two new properties. First, the error ε decreases with a more informative prior on $\theta_{s,*}$ (Section 4). In frequentist bounds, the prior plays the role of a regularizer, unrelated to the estimated model parameter, and thus cannot capture this effect. Second, we show that the hierarchical model in Figure 1 can improve statistical efficiency in multi-task bandits (Section 5). Our bounds and analyses are motivated by Bayes regret bounds in bandits (Russo & Van Roy, 2014; Lu & Van Roy, 2019; Kveton et al., 2021; Atsidakou et al., 2022; Hong et al., 2022b;a), which have similar properties and improve upon their frequentist counterparts similarly (Abbasi-Yadkori et al., 2011; Agrawal & Goyal, 2013).

3. Algorithm

Prior works in off-policy bandit and reinforcement learning often design pessimistic lower confidence bounds and then act according to them (Jin et al., 2021). We adopt the same design principle. Our LCBs satisfy $L_s(x, a) \leq r(x, a; \theta_{s,*})$ with a high probability for task parameter $\theta_{s,*} \mid \mathcal{D}$, jointly over all tasks s, contexts x, and actions a. Specifically, we define them as $L_s(x, a) = \hat{r}_s(x, a) - c_s(x, a)$, where

$$\begin{aligned} \hat{r}_s(x, a) &= \mathbb{E}\left[r(x, a; \theta_{s,*}) \,\middle|\, \mathcal{D}\right] , \\ c_s(x, a) &= \alpha \sqrt{\mathrm{var}\left[r(x, a; \theta_{s,*}) \,\middle|\, \mathcal{D}\right]} , \end{aligned} \tag{3}$$

are the estimated mean reward and its confidence interval width, respectively; and $\alpha > 0$ is a tunable parameter.

Linear models are an important class of contextual bandit models (Dani et al., 2008; Abbasi-Yadkori et al., 2011; Li et al., 2010) and we also consider them here. Specifically, we assume that $r(x, a; \theta_{s,*}) = \phi(x, a)^\top \theta_{s,*}$ for each task s, where $\theta_{s,*}$ is the task parameter and $\phi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$ is some *feature extractor*. Under this assumption, we can write (3) using the posterior mean and covariance of $\theta_{s,*}$ as

$$\begin{aligned} \hat{r}_s(x, a) &= \phi(x, a)^\top \mathbb{E}\left[\theta_{s,*} \,\middle|\, \mathcal{D}\right] , \\ c_s(x, a) &= \alpha \sqrt{\phi(x, a)^\top \mathrm{cov}\left[\theta_{s,*} \,\middle|\, \mathcal{D}\right] \phi(x, a)} . \end{aligned} \tag{4}$$

The above decomposition is desirable because it separates the posterior of the task parameter from context.

The rest of this section is organized as follows. We derive the mean reward estimate and its confidence interval width for a general model in Section 3.1. We instantiate these in a linear Gaussian model in Section 3.2 and then discuss the resulting algorithm in Section 3.3. Alternative algorithm designs are discussed in Section 3.4.

3.1. Hierarchical Pessimism

In any task s, the mean $\mathbb{E}\left[\theta_{s,*} \,\middle|\, \mathcal{D}\right]$ in (4) can be estimated hierarchically as follows. Let $\mathcal{D}_s$ be the subset of dataset $\mathcal{D}$ corresponding to task s, where $S_t = s$. Recall that μ_* is the hyper-parameter in Figure 1. Then, by the law of total expectation,

$$\begin{aligned} \mathbb{E}\left[\theta_{s,*} \,\middle|\, \mathcal{D}\right] &= \mathbb{E}\left[\mathbb{E}\left[\theta_{s,*} \,\middle|\, \mu_*, \mathcal{D}\right] \,\middle|\, \mathcal{D}\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[\theta_{s,*} \,\middle|\, \mu_*, \mathcal{D}_s\right] \,\middle|\, \mathcal{D}\right] . \end{aligned} \tag{5}$$

The second equality holds since conditioning on μ_* makes $\theta_{s,*}$ independent of $\mathcal{D} \setminus \mathcal{D}_s$. Our decomposition is motivated by the observation that estimating each $\mathbb{E}\left[\theta_{s,*} \,\middle|\, \mu_*, \mathcal{D}_s\right]$ is an easier problem than estimating $\mathbb{E}\left[\theta_{s,*} \,\middle|\, \mathcal{D}\right]$, since all observations in $\mathcal{D}_s$ are from a single task s. The information sharing between the tasks is still captured by μ_*, which is learned from the entire logged dataset $\mathcal{D}$.

Similarly, the covariance $\mathrm{cov}\left[\theta_{s,*} \,\middle|\, \mathcal{D}\right]$ in (4) can be decomposed using the law of total covariance,

$$\begin{aligned} &\mathrm{cov}\left[\theta_{s,*} \,\middle|\, \mathcal{D}\right] \\ &= \mathbb{E}\left[\mathrm{cov}\left[\theta_{s,*} \,\middle|\, \mu_*, \mathcal{D}\right] \,\middle|\, \mathcal{D}\right] + \mathrm{cov}\left[\mathbb{E}\left[\theta_{s,*} \,\middle|\, \mu_*, \mathcal{D}\right] \,\middle|\, \mathcal{D}\right] \\ &= \mathbb{E}\left[\mathrm{cov}\left[\theta_{s,*} \,\middle|\, \mu_*, \mathcal{D}_s\right] \,\middle|\, \mathcal{D}\right] + \mathrm{cov}\left[\mathbb{E}\left[\theta_{s,*} \,\middle|\, \mu_*, \mathcal{D}_s\right] \,\middle|\, \mathcal{D}\right] . \end{aligned} \tag{6}$$

Again, the second equality holds since conditioning on μ_* makes $\theta_{s,*}$ independent of $\mathcal{D} \setminus \mathcal{D}_s$. Note that (6) comprises two interpretable terms. The first captures the uncertainty of $\theta_{s,*}$ conditioned on μ_*, whereas the second captures the uncertainty in μ_*. This decomposes two sources of uncertainty in our problem, and is a powerful tool for structured uncertainty estimation (Hong et al., 2022a).

Now we plug (5) and (6) into (4), and get

$$\begin{aligned} \hat{r}_s(x, a) &= \phi(x, a)^\top \mathbb{E}\left[\mathbb{E}\left[\theta_{s,*} \,\middle|\, \mu_*, \mathcal{D}_s\right] \,\middle|\, \mathcal{D}\right] , \\ c_s(x, a) &= \alpha \sqrt{\phi(x, a)^\top \hat{\Sigma}_s \phi(x, a)} , \end{aligned}$$

where $\hat{\Sigma}_s =$

$$\mathbb{E}\left[\mathrm{cov}\left[\theta_{s,*} \,\middle|\, \mu_*, \mathcal{D}_s\right] \,\middle|\, \mathcal{D}\right] + \mathrm{cov}\left[\mathbb{E}\left[\theta_{s,*} \,\middle|\, \mu_*, \mathcal{D}_s\right] \,\middle|\, \mathcal{D}\right] .$$

With this in mind, we propose a general algorithm for hierarchical off-policy optimization, which we call `HierOPO`. Its pseudo-code is showed in Algorithm 1.

3.2. Hierarchical Gaussian Pessimism

The computation of (5) and (6) requires integrating out the hyper-parameter μ_* and task parameter $\theta_{s,*}$. This is generally impossible in a closed form, although many powerful approximations exist (Doucet et al., 2001). In this section, we leverage the conjugacy of a Gaussian hyper-prior, task priors, and reward distributions to obtain closed-form estimates of all model parameters. In this case, `HierOPO` can

Algorithm 1 HierOPO: Hierarchical off-policy optimization.

```
Input: Dataset $\mathcal{D}$
for $s \in \mathcal{S}, x \in \mathcal{X}$ do
    for $a \in \mathcal{A}$ do
        Compute $\hat{r}_s(x, a)$ and $c_s(x, a)$ (Section 3.1)
        $L_s(x, a) \leftarrow \hat{r}_s(x, a) - c_s(x, a)$
    $\hat{\pi}_s(x) \leftarrow \arg\max_{a \in \mathcal{A}} L_s(x, a)$
Output: $\hat{\pi} \leftarrow (\hat{\pi}_s)_{s \in \mathcal{S}}$
```

be implemented exactly and efficiently. We also analyze it under these assumptions (Section 5).

In particular, we consider a linear Gaussian model with the hyper-prior $Q = \mathcal{N}(\mu_q, \Sigma_q)$ and the task prior $P(\cdot \mid \mu_*) = \mathcal{N}(\cdot; \mu_*, \Sigma_0)$. The mean vector $\mu_q \in \mathbb{R}^d$, as well as both the covariance matrices $\Sigma_q, \Sigma_0 \in \mathbb{R}^{d \times d}$, are assumed to be known. The reward distribution of action a in context x is $\mathcal{N}(\phi(x, a)^\top \theta_{s,*}, \sigma^2)$, where ϕ is a feature extractor and $\sigma > 0$ is a known reward noise. It follows that the mean reward is linear in features.

To derive (5) and (6), we start with understanding posterior distributions of $\theta_{s,*}$ and μ_*. Specifically, conditioning in Gaussian graphical models preserves Gaussianity, and thus $\theta_{s,*} \mid \mu_*, \mathcal{D}_s \sim \mathcal{N}(\tilde{\mu}_s, \tilde{\Sigma}_s)$ for some $\tilde{\mu}_s$ and $\tilde{\Sigma}_s$. Using the structure of our model (Figure 1), we further note that this is a standard linear model posterior with a Gaussian prior $\mathcal{N}(\mu_*, \Sigma_0)$, where

$$\begin{aligned} \tilde{\mu}_s &= \mathbb{E}\left[\theta_{s,*} \,\middle|\, \mu_*, \mathcal{D}_s\right] = \tilde{\Sigma}_s(\Sigma_0^{-1}\mu_* + B_s)\,, \\ \tilde{\Sigma}_s &= \operatorname{cov}\left[\theta_{s,*} \,\middle|\, \mu_*, \mathcal{D}_s\right] = (\Sigma_0^{-1} + G_s)^{-1}\,, \end{aligned} \tag{7}$$

and the statistics

$$\begin{aligned} B_s &= \sigma^{-2} \sum_{t=1}^{n} \mathbb{1}\{S_t = s\}\, \phi(X_t, A_t) Y_t\,, \\ G_s &= \sigma^{-2} \sum_{t=1}^{n} \mathbb{1}\{S_t = s\}\, \phi(X_t, A_t) \phi(X_t, A_t)^\top\,, \end{aligned} \tag{8}$$

are computed using the subset $\mathcal{D}_s$ of the logged dataset $\mathcal{D}$.

The posterior of the hyper-parameter $\mu_* \mid \mathcal{D}$, known as the hyper-posterior, also has a closed-form $\mathcal{N}(\bar{\mu}, \bar{\Sigma})$ (Section 4.2 of Hong et al. 2022b), where

$$\begin{aligned} \bar{\mu} &= \mathbb{E}\left[\mu_* \,\middle|\, \mathcal{D}\right] \\ &= \bar{\Sigma}\Big(\Sigma_q^{-1}\mu_q + \sum_{s \in \mathcal{S}} (\Sigma_0 + G_s^{-1})^{-1} G_s^{-1} B_s\Big)\,, \\ \bar{\Sigma} &= \operatorname{cov}\left[\mu_* \,\middle|\, \mathcal{D}\right] = \Big(\Sigma_q^{-1} + \sum_{s \in \mathcal{S}} (\Sigma_0 + G_s^{-1})^{-1}\Big)^{-1}. \end{aligned} \tag{9}$$

One way of interpreting (9) is as a multivariate Gaussian posterior where each task is a single observation. The observation of task s is the least-squares estimate of $\theta_{s,*}$ from task s, given by $G_s^{-1} B_s$, with covariance $\Sigma_0 + G_s^{-1}$. The tasks with more observations affect the estimate $\bar{\mu}$ more, since their G_s^{-1} approaches a zero matrix, and as a result $\Sigma_0 + G_s^{-1} \to \Sigma_0$. This uncertainty is intrinsic, since even $\theta_{s,*}$ is a noisy observation of μ_*.

To complete our derivations, we only need to substitute (7) and (9) into (5) and (6). The posterior mean of $\theta_{s,*}$ is

$$\begin{aligned} \mathbb{E}\left[\mathbb{E}\left[\theta_{s,*} \,\middle|\, \mu_*, \mathcal{D}_s\right] \,\middle|\, \mathcal{D}\right] &= \mathbb{E}\left[\tilde{\Sigma}_s(\Sigma_0^{-1}\mu_* + B_s) \,\middle|\, \mathcal{D}\right] \\ &= \tilde{\Sigma}_s(\Sigma_0^{-1}\mathbb{E}\left[\mu_* \,\middle|\, \mathcal{D}\right] + B_s) \\ &= \tilde{\Sigma}_s(\Sigma_0^{-1}\bar{\mu} + B_s)\,, \end{aligned}$$

where we simply combine (7) and (9). Similarly, the posterior covariance of $\theta_{s,*}$ requires computing

$$\begin{aligned} \mathbb{E}\left[\operatorname{cov}\left[\theta_{s,*} \,\middle|\, \mu_*, \mathcal{D}_s\right] \,\middle|\, \mathcal{D}\right] &= \mathbb{E}\left[\tilde{\Sigma}_s \,\middle|\, \mathcal{D}\right] = \tilde{\Sigma}_s\,, \\ \operatorname{cov}\left[\mathbb{E}\left[\theta_{s,*} \,\middle|\, \mu_*, \mathcal{D}_s\right] \,\middle|\, \mathcal{D}\right] &= \operatorname{cov}\left[\tilde{\Sigma}_s(\Sigma_0^{-1}\mu_* + B_s) \,\middle|\, \mathcal{D}\right] \\ &= \operatorname{cov}\left[\tilde{\Sigma}_s\Sigma_0^{-1}\mu_* \,\middle|\, \mathcal{D}\right] \\ &= \tilde{\Sigma}_s\Sigma_0^{-1}\bar{\Sigma}\Sigma_0^{-1}\tilde{\Sigma}_s\,. \end{aligned}$$

Finally, the estimated mean reward and its confidence interval width are given by

$$\begin{aligned} \hat{r}_s(x, a) &= \phi(x, a)^\top \tilde{\Sigma}_s(\Sigma_0^{-1}\bar{\mu} + B_s)\,, \\ c_s(x, a) &= \alpha\sqrt{\phi(x, a)^\top \hat{\Sigma}_s \phi(x, a)}\,, \end{aligned} \tag{10}$$

where $\hat{\Sigma}_s = \tilde{\Sigma}_s + \tilde{\Sigma}_s\Sigma_0^{-1}\bar{\Sigma}\Sigma_0^{-1}\tilde{\Sigma}_s$. The posterior covariance $\hat{\Sigma}_s$ is tractable and has the following desirable properties. First, the hyper-parameter uncertainty only affects the second term through $\bar{\Sigma}$. Moreover, since $\tilde{\Sigma}_s$ appears in both terms, $\hat{\Sigma}_s$ decreases with more observations from task s.

3.3. Gaussian HierOPO

Now we plug the estimated mean reward and its confidence interval width in (10) into HierOPO. The resulting method is a form of hierarchical regression of the hyper-parameter and task parameters. The task parameter $\theta_{s,*}$ is estimated using $\tilde{\Sigma}_s(\Sigma_0^{-1}\bar{\mu} + B_s)$ in (10) and the hyper-parameter μ_* is estimated using $\bar{\mu}$ in (9). Since HierOPO is a variant of hierarchical regression, it is fairly general and we expect it to perform well beyond our assumptions in the algorithm design, such as when the reward noise is sub-Gaussian.

To simplify the presentation of HierOPO, we assume that $\mathcal{X}$ and $\mathcal{A}$ are finite. However, since HierOPO is based on a hierarchical regression of the task parameter in (10) and the hyper-parameter in (9), this assumption is not necessary. In fact, the mean reward estimate and its confidence interval at any feature vector $\phi(x, a)$ can be computed as described in (10). Our error bounds in Section 5 are also independent of the number of contexts or actions. Finally, when the action space cannot be enumerated, it may be hard to compute the most pessimistic action $\hat{\pi}_s(x) = \arg\max_{a \in \mathcal{A}} L_s(x, a)$ in context x. In this case, our algorithm and analysis can be extended to any maximization oracle with a fixed approximation ratio.

The computations in (10) rely on matrix inversions, whose computational cost is $O(d^3)$ for d features. Note that this is only needed for the exact implementation. Approximate inference, which trades off the computational cost for accuracy (Doucet et al., 2001), is possible. Any approach for Gaussian graphical models would apply in our setting.

3.4. Alternative Designs

A natural question to ask is how the hierarchy helps with improving pessimistic reward estimates. To answer it, we compare `HierOPO` to two baselines, based on pessimistic least-squares estimators (Li et al., 2022) that do not model our structure. The first one is unrealistic because it assumes that μ_* is known. We call it `OracleOPO`. Here the posterior mean reward and its confidence interval width are

$$\begin{aligned} \hat{r}_s(x, a) &= \phi(x, a)^\top \tilde{\Sigma}_s (\Sigma_0^{-1} \mu_* + B_s)\,, \\ c_s(x, a) &= \alpha \sqrt{\phi(x, a)^\top \tilde{\Sigma}_s \phi(x, a)}\,. \end{aligned} \tag{11}$$

This is an improvement of (10) in two aspects. First, the estimate $\bar{\mu}$ of μ_* is replaced with the actual μ_*. Second, the confidence interval width is provably narrower because

$$\tilde{\Sigma}_s \preceq \tilde{\Sigma}_s + \tilde{\Sigma}_s \Sigma_0^{-1} \bar{\Sigma} \Sigma_0^{-1} \tilde{\Sigma}_s\,.$$

The second method does not model the hyper-parameter μ_*. Instead, its uncertainty is incorporated into that of modeled $\theta_{s,*}$. This is achieved by replacing Σ_0 in (11) with $\Sigma_q + \Sigma_0$, and μ_* with μ_q. We call the method `FlatOPO`. Its posterior mean reward and confidence interval width are

$$\begin{aligned} \hat{r}_s(x, a) &= \phi(x, a)^\top \dot{\Sigma}_s ((\Sigma_q + \Sigma_0)^{-1} \mu_q + B_s)\,, \\ c_s(x, a) &= \alpha \sqrt{\phi(x, a)^\top \dot{\Sigma}_s \phi(x, a)}\,, \end{aligned}$$

where

$$\dot{\Sigma}_s = ((\Sigma_q + \Sigma_0)^{-1} + G_s)^{-1}\,.$$

In comparison to (10), this method is worse in two aspects. First, the prior mean μ_q of μ_* is used instead of its estimate $\bar{\mu}$. Second, as the number of tasks m increases, we expect $\lambda_1(\bar{\Sigma}) \to 0$ and then

$$\dot{\Sigma}_s \succeq \tilde{\Sigma}_s \Sigma_0^{-1} \bar{\Sigma} \Sigma_0^{-1} \tilde{\Sigma}_s + \tilde{\Sigma}_s\,.$$

Therefore, our approach should be more statistically efficient, which we prove formally in Section 5.

Finally, note that optimistic methods, such as posterior sampling (Thompson, 1933; Russo et al., 2018) and `BayesUCB` (Kaufmann et al., 2012), cannot be used in our setting. In fact, optimism is harmful because it leads to taking highly uncertain actions whose uncertainty is not reduced, since there are no additional observations from the environment. In this case, pessimism and robustness are desired, and these are our main design principles.

4. Single-Task Analysis

To illustrate Bayesian error bounds, we start with a classic contextual bandit parameterized by $\theta_* \in \mathbb{R}^d$. The mean reward of action $a \in \mathcal{A}$ in context $x \in \mathcal{X}$ under parameter $\theta \in \mathbb{R}^d$ is $r(x, a; \theta) = \phi(x, a)^\top \theta$. We assume that $\theta_* \sim \mathcal{N}(\theta_0, \Sigma_0)$ and the reward noise is $\mathcal{N}(0, \sigma^2)$. This model is identical to a single task in Section 3.2.

The logged dataset is $\mathcal{D} = \{(X_t, A_t, Y_t)\}_{t=1}^n$, the LCB is $L(x, a) = \hat{r}(x, a) - c(x, a)$, and the pessimistic policy is $\hat{\pi}(x) = \arg\max_{a \in \mathcal{A}} L(x, a)$. Following the same reasoning as in the derivation of (7), the estimated mean reward and its confidence interval width are

$$\begin{aligned} \hat{r}(x, a) &= \phi(x, a)^\top \hat{\Sigma} (\Sigma_0^{-1} \theta_0 + B)\,, \\ c(x, a) &= \alpha \sqrt{\phi(x, a)^\top \hat{\Sigma} \phi(x, a)}\,, \end{aligned}$$

where

$$\begin{aligned} B &= \sigma^{-2} \sum_{t=1}^n \phi(X_t, A_t) Y_t\,, \\ \hat{\Sigma} &= (\Sigma_0^{-1} + G)^{-1}\,, \\ G &= \sigma^{-2} \sum_{t=1}^n \phi(X_t, A_t) \phi(X_t, A_t)^\top\,. \end{aligned} \tag{12}$$

As in Section 2, the value of policy $\pi \in \Pi$ under model parameter θ_* is $V(\pi; \theta_*) = \mathbb{E}\left[r(X, \pi(X); \theta_*)\right]$. The optimal policy is $\pi_* = \arg\max_{\pi \in \Pi} V(\pi; \theta_*)$. The quality of policy $\hat{\pi}$ is measured by an (ε, δ) bound

$$\mathbb{P}\left(V(\hat{\pi}; \theta_*) \geq V(\pi_*; \theta_*) - \varepsilon \,|\, \mathcal{D}\right) \geq 1 - \delta\,. \tag{13}$$

A better policy has a lower $\varepsilon > 0$ at a fixed $\delta > 0$. Now we are ready to proceed with the analysis.

4.1. Bayesian Error Bound

We start with assumptions. First, we assume that the length of feature vectors is bounded.

Assumption 1. *For any context $x \in \mathcal{X}$ and action $a \in \mathcal{A}$, the feature vector satisfies $\|\phi(x, a)\|_2 \leq 1$.*

This assumption is standard and without loss of generality. Second, we assume that the dataset $\mathcal{D}$ is "well-explored" (Swaminathan et al., 2017; Jin et al., 2021).

Assumption 2. *Take G in (12) and let*

$$G_* = \mathbb{E}\left[\phi(X, \pi_*(X))\phi(X, \pi_*(X))^\top \,\middle|\, \theta_*\right] .$$

Then there exists $\gamma > 0$ such that $G \succeq \gamma\sigma^{-2} n G_*$ holds for any θ_*.

Assumption 2 relates the logging policy π_0, which induces G, to the optimal policy π_*, which induces $\sigma^{-2} n G_*$. It can be loosely interpreted as follows. As n increases,

$$G \to \sigma^{-2} n \, \mathbb{E}\left[\phi(X, \pi_0(X))\phi(X, \pi_0(X))^\top \,\middle|\, \theta_*\right] ,$$

and γ becomes the maximum ratio between probabilities of taking actions by π_* and π_0 in any direction. Therefore, the assumption measures the degree of overlap between π_* and π_0. It also relates a finite-sample G to the expected G_*. Therefore, we do not need to reason about a finite-sample behavior of G in our analysis.

Note that Assumption 2 holds for $\gamma = 0$. Unfortunately, this setting would negate the desired scaling with sample size n in our error bounds and is impractical. The higher the value of γ, the more π_* and π_0 are similar. When π_0 is uniform, we obtain $\gamma = \Omega(1/d)$ for large n. Now we state our main claim for the single-task setting.

Theorem 1. *Fix dataset $\mathcal{D}$ and choose any $\gamma > 0$ such that Assumption 2 holds. Let $\hat{\pi}(x) = \arg\max_{a \in \mathcal{A}} L(x, a)$. Then for any $\delta \in (0, e^{-1}]$, the error in (13) is*

$$\varepsilon = \underbrace{\sqrt{5d \log(1/\delta)}}_{\alpha} \sqrt{\frac{4d}{\lambda_d(\Sigma_0^{-1}) + \gamma\sigma^{-2} n}} .$$

Proof. The claim is proved in Appendix A in three steps. First, we establish that $c(x, a)$ is a high-probability confidence interval width for $\alpha = \sqrt{5d \log(1/\delta)}$. Second, we show that the suboptimality of policy $\hat{\pi}$ can be bounded by $\mathbb{E}\left[c(X, \pi_*(X))\right]$ for any parameter θ_*. Third, we combine $c(x, a)$ with Assumption 2, and relate the logging policy π_0 that induces $c(x, a)$ with the optimal policy π_*. □

4.2. Frequentist Error Bound

To understand the benefit of a Bayesian analysis, we compare Theorem 1 to a frequentist bound. The main difference in the frequentist bound is that θ_* is fixed. Thus a natural counterpart of (13) is

$$V(\hat{\pi}; \theta_*) \geq V(\pi_*; \theta_*) - \varepsilon ,$$

which holds with probability at least $1 - \delta$ for an unknown but fixed θ_*. Under the assumptions that $\|\theta_*\|_2 \leq \kappa$, and that $(Y_t)_{t=1}^n$ are independent σ^2-sub-Gaussian rewards, we get a similar bound to Theorem 1, which is stated in Theorem 5 (Appendix B). The main difference is that

$$\alpha = 2\sqrt{2d(\log(1/\delta) + b)} + \kappa \lambda_d^{-\frac{1}{2}}(\Sigma_0) , \qquad (14)$$

where Σ_0 should be viewed as a regularization parameter instead of the prior covariance. Before we discuss α, we discuss two key differences in how the bounds are stated. First, the frequentist bound holds for any model parameter θ_* such that $\|\theta_*\|_2 \leq \kappa$. Therefore, it is arguably stronger than the Bayesian bound in Theorem 5. This is analogous to differences in Bayesian and frequentist cumulative regret bounds (Russo & Van Roy, 2014). Second, because both bounds depend on γ in Assumption 2, which depends on θ_*, we make an assumption that π_0 is uniform. In this case, $\gamma = \Omega(1/d)$ for any θ_*. Therefore, γ has no impact on the next discussion and we may treat it as a constant.

Under the above assumptions, the only major difference in the bounds is the term $\kappa \lambda_d^{-\frac{1}{2}}(\Sigma_0)$ in (14). This term can have a major effect. For instance, suppose that $\Sigma_0 = I_d / n$ in (12). From a Bayesian viewpoint, this corresponds to a very informative prior with width $\sqrt{1/n}$, and the Bayesian bound in Theorem 1 is $\tilde{O}(dn^{-\frac{1}{2}})$. From a frequentist point of view, this amounts to $O(n)$ regularization, and the frequentist bound becomes $\tilde{O}(dn^{-\frac{1}{2}} + d^{\frac{1}{2}})$. As $n \to \infty$, we get that the Bayesian bound can be arbitrarily better.

5. Multi-Task Analysis

Now we study our multi-task setting, where the estimated mean reward and its confidence interval width are defined in (10). Similarly to Section 4, our analysis is Bayesian. We derive an error bound for a single task and discuss how to extend it to other bounds, such as for all tasks, later.

5.1. Bayesian Error Bound

To derive the bound in (2), we make similar assumptions to Section 4. First, we assume that the length of feature vectors is bounded (Assumption 1). Second, we assume that the dataset $\mathcal{D}$ is "well-explored" for all tasks.

Assumption 3. *Take G_s in (8) and let n_s be the number of interactions with task s. Let*

$$G_{s,*} = \mathbb{E}\left[\phi(X, \pi_{s,*}(X))\phi(X, \pi_{s,*}(X))^\top \,\middle|\, \theta_{s,*}\right] .$$

Then there exists $\gamma > 0$ such that $G_s \succeq \gamma\sigma^{-2} n_s G_{s,*}$ holds for any $\theta_{s,*}$ in any task $s \in \mathcal{S}$.

This is essentially Assumption 2 applied to all tasks. For a uniform logging policy, $\gamma = \Omega(1/d)$ when n_s is large for all tasks $s \in \mathcal{S}$. So the assumption is not very strong. Our main technical result is presented below.

Theorem 2. *Fix dataset $\mathcal{D}$ and choose any $\gamma > 0$ such that Assumption 3 holds. Take policy $\hat{\pi}$ computed by* `HierOPO` *and let $\alpha = \sqrt{5d \log(1/\delta)}$. Then for any $\delta \in (0, e^{-1}]$, the* *error in* (2) *is*

$$\varepsilon = \underbrace{\alpha\sqrt{\frac{4d}{\lambda_d(\Sigma_0^{-1}) + \gamma\sigma^{-2}n_s}}}_{\textit{Task term}} + \underbrace{\alpha\sqrt{\frac{4d}{\lambda_d(\Sigma_q^{-1}) + \sum_{z \in \mathcal{S}} \frac{1}{\lambda_1(\Sigma_0) + \gamma^{-1}\sigma^2\lambda_1(G_{z,*}^{-1})n_z^{-1}}}}}_{\textit{Hyper-parameter term}} .$$

Moreover, suppose that $\phi(x, a)$ *has at most one non-zero entry for any* $x \in \mathcal{X}$ *and* $a \in \mathcal{A}$*, and that both* Σ_q *and* Σ_0 *are diagonal. Then* $\lambda_1(G_{z,*}^{-1}) \geq 1$*.*

Proof. The claim is proved in Appendix C, following the same three steps as in the proof of Theorem 1. The main difference is in the definitions of $c(x, a)$ and policies, and that Assumption 3 is used instead of Assumption 2. This shows the generality of our proof technique and indicates that it may apply to other graphical models. □

5.2. Discussion

Our error bound is Bayesian and proved for a distribution of the task parameter $\theta_{s,*} \mid \mathcal{D}$. The bound has two terms. The former captures the error in estimating the task parameter $\theta_{s,*}$ if the hyper-parameter μ_* was known. It is similar to Theorem 1 and hence we call it the *task term*. The latter term captures the error in estimating μ_* and hence we call it the *hyper-parameter term*. We discuss each term next.

The task term depends on all quantities of interest in an expected manner. It is $O(d\sqrt{\log(1/\delta)/n_s})$, where d is the number of features, n_s is the number of observations, and δ is the probability that the bound fails. This dependence is standard in confidence intervals for linear models with an infinite number of contexts (Abbasi-Yadkori et al., 2011; Agrawal & Goyal, 2013; Abeille & Lazaric, 2017). Since $\lambda_d(\Sigma_0^{-1})$ can be viewed as the minimum number of prior pseudo-observations in any direction in $\mathbb{R}^d$, the task term decreases with a more informative prior. Finally, the task term decreases when the observation noise σ decreases, and the similarity of the logging and optimal policies γ increases (Assumption 3).

The hyper-parameter term mimics scaling of the task term at the hyper-parameter level. In particular, the minimum number of prior pseudo-observations in any direction in $\mathbb{R}^d$ becomes $\lambda_d(\Sigma_q^{-1})$ and each task becomes an observation, which is reflected by the sum over all tasks z. The hyper-parameter term decreases as the number of observations n_z in any task z increases, the maximum width of the task prior $\sqrt{\lambda_1(\Sigma_0)}$ decreases, reward noise σ decreases, and the similarity of logging and optimal policies γ increases.

To show that our bound captures the problem structure, we compare it to two baselines in Section 3.4: `OracleOPO` and `FlatOPO`. `OracleOPO` knows μ_* and has more information than `HierOPO`. Its error can be bounded by Theorem 1 and is always lower than that of `HierOPO`, because the bound corresponds to the first term in Theorem 2. On the other hand, `FlatOPO` solves each task independently. Its error can be bounded using Theorem 1 where the task covariance Σ_0 is replaced by $\Sigma_q + \Sigma_0$, to account for the additional uncertainty due to unknown μ_*. The resulting error bound becomes

$$\alpha\sqrt{\frac{4d}{\lambda_d((\Sigma_q + \Sigma_0)^{-1}) + \gamma\sigma^{-2}n_s}}$$

and is always higher than the task term in Theorem 2. As the number of tasks increases, the hyper-parameter term in Theorem 2 goes to zero, and the error bound of `HierOPO` would be provably lower.

Finally, we would like to comment on the second claim in Theorem 2, which results in a tighter bound. This claim is proved under additional assumptions that are satisfied by a multi-armed bandit, for instance.

5.3. Extensions

The error bound in Theorem 2 is derived for a fixed task $s \in \mathcal{S}$. This decision was taken deliberately because other bounds can be easily derived from it. For instance, to get a bound for all tasks, we only need a union bound over all $\theta_{s,*} \mid \mathcal{D}$. As a result, Theorem 2 holds for all $s \in \mathcal{S}$ with probability at least $1 - m\delta$.

The bound in Theorem 2 also holds for a new task sampled from the task prior. This is because the hyper-parameter estimation in (9), which affects the hyper-parameter term in Theorem 2, separates all tasks from the evaluated one.

6. Related Work

Off-policy optimization. In off-policy optimization, data collected by a deployed policy are used to learn improved policies offline (Li et al., 2010), without a direct interaction with the environment. Off-policy estimation and optimization can be model-free or model-based. Many model-free methods are based on inverse propensity scores (IPS) (Horvitz & Thompson, 1952; Ionides, 2008; Strehl et al., 2010; Swaminathan & Joachims, 2015). These methods have a low bias and high variance, unless corrected. Model-based methods estimate a reward model for context-action pairs, which is then used to find an optimal policy (Bottou et al., 2013; Dudik et al., 2014). These approaches tend to have a high bias and low variance. Doubly-robust estimation (Robins et al., 1994; Dudik et al., 2014) is used frequently to combine model-based and model-free methods. We take

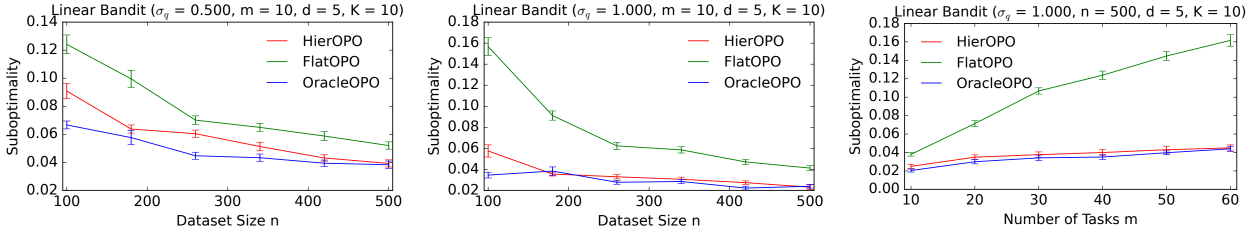

Figure 2. Evaluation of off-policy algorithms on the synthetic multi-task bandit problem. In the left and middle plots, we vary the dataset size n for small $\sigma_q = 0.5$ and large $\sigma_q = 1.0$, respectively. In the right plot, we vary the number of tasks m.

a model-based approach in this work.

Offline reinforcement learning. Pessimism has been studied extensively in offline reinforcement learning (Buckman et al., 2021; Jin et al., 2021). Specifically, Jin et al. (2021) showed that pessimistic value iteration is minimax optimal in linear Markov decision processes (MDPs). Multi-task offline reinforcement learning was also studied by Lazaric & Ghavamzadeh (2010). This paper applied expectation-maximization to solve the problem but did not prove any error bounds. In comparison, we consider a simpler setting of contextual bandits and prove error bounds that show improvements due to using the multi-task structure. These are the first error bounds of its kind.

Online learning. Off-policy methods learn from data collected by another policy. In contrast, online methods learn from data that they collect, and need to balance exploration and exploitation to attain low regret in the long term. Two popular exploration techniques are upper confidence bounds (UCBs) (Auer et al., 2002) and posterior sampling (Thompson, 1933), and both have been studied extensively in linear models (Dani et al., 2008; Abbasi-Yadkori et al., 2011; Chu et al., 2011; Agrawal & Goyal, 2013). Bandit algorithms for hierarchical models have also been studied extensively (Bastani et al., 2019; Kveton et al., 2021; Basu et al., 2021; Simchowitz et al., 2021; Wan et al., 2021; Hong et al., 2022b; Peleg et al., 2022; Wan et al., 2022). Perhaps surprisingly, all of these are based on posterior sampling. Our marginal posterior derivations in (10) can used to derive UCBs for this setting. Specifically, $U_s(x, a) = \hat{r}_s(x, a) + c_s(x, a)$ is an upper confidence bound on the mean reward of action a in context x and task s.

7. Experiments

In this section, we empirically compare `HierOPO` to baselines `OracleOPO` and `FlatOPO` (Section 3.4). All methods are implemented as described in Section 3. We set $\alpha = 0.1$, which led to good performance in our initial experiments. The goal of our experiments is to show that hierarchy can greatly improve the statistical efficiency of off-policy algorithms. We include an additional experiment in Appendix D, where `HierOPO` is applied to a challenging image classification task with deep neural networks.

7.1. Synthetic Problem

Our first experiment is with a synthetic multi-task bandit, with $d = 5$ features and $K = 10$ actions. For each action $a \in \mathcal{A}$ and interaction $t \in [n]$, we sample a feature vector uniformly at random from $[-0.5, 0.5]^d$. The hierarchy is defined as follows. The hyper-prior is $\mathcal{N}(\mathbf{0}_d, \Sigma_q)$, where $\Sigma_q = \sigma_q^2 I_d$ is its covariance. The task covariance is $\Sigma_0 = \sigma_0^2 I_d$. We experiment with $\sigma_q \in \{0.5, 1\}$ and $\sigma_0 = 0.5$. We expect the hierarchy to be more beneficial when $\sigma_q > \sigma_0$, since the uncertainty of the hyper-parameter is higher and it is more valuable to learn it. The reward distribution of task s is $\mathcal{N}(\phi(x, a)^\top \theta_{s,*}, \sigma^2)$ with noise $\sigma = 0.5$.

Our results are averaged over multiple runs. At the beginning of each run, the hyper-parameter μ_* is sampled from the hyper-prior $\mathcal{N}(\mathbf{0}_d, \Sigma_q)$. After that, each task parameter is sampled i.i.d. as $\theta_{s,*} \sim \mathcal{N}(\mu_*, \Sigma_0)$. The logged dataset $\mathcal{D}$ is generated as follows. In each interaction $t \in [n]$, we randomly select a task, take a random action, and record its stochastic reward. The learned policies $\hat{\pi}_s$ are evaluated on the same problem that generated $\mathcal{D}$. The evaluation criterion is the *suboptimality* of learned policies averaged over the tasks, which we define as

$$\frac{1}{m} \sum_{s \in \mathcal{S}} V(\pi_{s,*}; \theta_{s,*}) - V(\hat{\pi}_s; \theta_{s,*}) \,.$$

The policy values and the optimal policy are estimated on another logged dataset of size 10 000. In our experiments, we vary either the logged dataset size n or the number of tasks m, while keeping the other fixed. The default settings are $m = 10$ tasks and logged dataset size $n = 500$.

In Figure 2, we show the mean and standard error of the suboptimality of each algorithm averaged over 30 random runs. As expected, `HierOPO` outperforms `FlatOPO` and is close to `OracleOPO`. The improvement is higher when the hyper-parameter uncertainty σ_q is higher. The difference between `HierOPO` and `FlatOPO` is the most noticeable in the limited data regime, where n is small or m is large. In both cases, the number of observations per task is small.

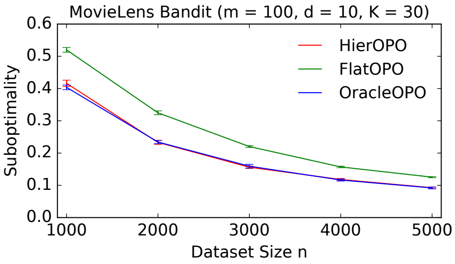

Figure 3. Evaluation of off-policy algorithms on a multi-user recommendation problem in Section 7.2.

7.2. Multi-User Recommendation

Now we consider a multi-user recommendation problem. The problem is simulated using the MovieLens 1M dataset (Lam & Herlocker, 2016), with 1 million ratings for 3 883 movies from 6 040 users. As a first step, we complete the sparse rating matrix M using alternating least squares (Davenport & Romberg, 2016) with rank $d = 10$. This rank is sufficiently high to have a low prediction error, but also low enough to prevent overfitting. The learned factorization is $M = UV^\top$. The i-th row of U, denoted by U_i, represents user i. The j-th row of V, denoted by V_j, represents movie j. The reward for recommending movie j to user i is sampled from $\mathcal{N}(V_j^\top U_i, \sigma^2)$. The reward noise $\sigma = 0.759$ is estimated from data. The feature vectors in each interaction are latent factors V_j of 30 randomly chosen movies.

To simulate similar users, we cluster user latent factors. In particular, we apply a *Gaussian mixture model (GMM)* with $k = 7$ clusters to rows of U. We choose the smallest value of k that yields well-separated clusters (Bishop, 2006). The hyper-prior parameters μ_q and Σ_q are set to the mean and covariance of all cluster centers, respectively. The cluster with most users represents tasks. We set μ_* and Σ_0 to its center and covariance estimated by the GMM, respectively. Thus all users in the cluster are related through μ_* and Σ_0. We want to stress that the GMM is only used to estimate the hyper-parameters of the hierarchical model. The task parameters are user latent factors U_i. This is to ensure that our setup is as realistic as possible.

The number of tasks is $m = 100$ and they are chosen randomly in each run. The dataset $\mathcal{D}$ is logged as follows. In each interaction $t \in [n]$, we randomly select a task, take a random action, and record its stochastic reward. The evaluation criteria are the same as in Section 7.1.

In Figure 3, we report the mean and standard error of the suboptimality of all algorithms averaged over 10 random runs. For all dataset sizes n, `HierOPO` performs very well: its suboptimality is close to that of `OracleOPO` and significantly lower than that of `FlatOPO`. This clearly shows the benefit of hierarchies for efficient off-policy learning. Also note that the hierarchy in this experiment is estimated from data. Therefore, it is misspecified; yet hugely beneficial.

8. Conclusions

In this work, we propose hierarchical off-policy optimization (`HierOPO`), a general off-policy algorithm for solving similar contextual bandit tasks related through a hierarchy. Our algorithm leverages the hierarchical structure to learn tighter, and hence more sample efficient, lower confidence bounds on the mean rewards of actions and acts according to them. We derive Bayesian error bounds for our policies, which become tighter with a more informative prior and demonstrate the benefit of hierarchies. Finally, we empirically validate the effectiveness of hierarchies on synthetic and real-world problems.

Our work is the first to propose a practical and analyzable algorithm for off-policy learning with hierarchical Bayesian models. As a result, there are many potential directions for future work. First, some applications require more complex graphical models than a two-level hierarchy with a single hyper-parameter (Hong et al., 2022a; Aouali et al., 2023). We believe that our methodology directly extends to these. Second, we believe that `HierOPO` and its analysis can be extended to reinforcement learning (Lazaric & Ghavamzadeh, 2010). Third, `HierOPO` is a model-based approach to off-policy learning (Section 6). Since model-based approaches tend to be biased, due to using a potentially misspecified model, it is important to develop model-free methods for multi-task off-policy learning.

Finally, our theory shows benefits of a Bayesian analysis over a frequentist one (Section 4), and also benefits of hierarchies (Section 5). We are not aware of matching lower bounds for our setting and these are not common in offline learning, unlike in bandits (Lattimore & Szepesvari, 2019). This leaves open the possibility that our bounds are loose. We believe that this is highly unlikely, since the bounds are derived using exact Gaussian posteriors.

References

Abbasi-Yadkori, Y., Pal, D., and Szepesvari, C. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24*, pp. 2312–2320, 2011.

Abeille, M. and Lazaric, A. Linear Thompson sampling revisited. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

Agrawal, S. and Goyal, N. Analysis of Thompson sampling for the multi-armed bandit problem. In *Proceeding of the 25th Annual Conference on Learning Theory*, pp. 39.1–39.26, 2012.

Agrawal, S. and Goyal, N. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 127–135, 2013.

Aouali, I., Kveton, B., and Katariya, S. Mixed-effect Thompson sampling. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, 2023.

Atsidakou, A., Katariya, S., Sanghavi, S., and Kveton, B. Bayesian fixed-budget best-arm identification. *CoRR*, abs/2211.08572, 2022. URL https://arxiv.org/abs/2211.08572.

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002.

Azar, M. G., Lazaric, A., and Brunskill, E. Sequential transfer in multi-armed bandit with finite set of models. In *Advances in Neural Information Processing Systems 26*, pp. 2220–2228, 2013.

Bastani, H., Simchi-Levi, D., and Zhu, R. Meta dynamic pricing: Transfer learning across experiments. *CoRR*, abs/1902.10918, 2019. URL https://arxiv.org/abs/1902.10918.

Basu, S., Kveton, B., Zaheer, M., and Szepesvari, C. No regrets for learning the prior in bandits. In *Advances in Neural Information Processing Systems 34*, 2021.

Bishop, C. *Pattern Recognition and Machine Learning*. Springer, New York, NY, 2006.

Bottou, L., Peters, J., Quinonero-Candela, J., Charles, D., Chickering, M., Portugaly, E., Ray, D., Simard, P., and Snelson, E. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14(101):3207–3260, 2013.

Boucheron, S., Lugosi, G., and Massart, P. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

Buckman, J., Gelada, C., and Bellemare, M. The importance of pessimism in fixed-dataset policy optimization. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

Cella, L., Lazaric, A., and Pontil, M. Meta-learning with stochastic linear bandits. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Chapelle, O. and Li, L. An empirical evaluation of Thompson sampling. In *Advances in Neural Information Processing Systems 24*, pp. 2249–2257, 2012.

Chu, W., Li, L., Reyzin, L., and Schapire, R. Contextual bandits with linear payoff functions. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pp. 208–214, 2011.

Dani, V., Hayes, T., and Kakade, S. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning Theory*, pp. 355–366, 2008.

Davenport, M. and Romberg, J. An overview of low-rank matrix recovery from incomplete observations. *IEEE Journal of Selected Topics in Signal Processing*, 10(4): 608–622, 2016.

Deshmukh, A. A., Dogan, U., and Scott, C. Multi-task learning for contextual bandits. In *Advances in Neural Information Processing Systems 30*, pp. 4848–4856, 2017.

Doucet, A., de Freitas, N., and Gordon, N. *Sequential Monte Carlo Methods in Practice*. Springer, New York, NY, 2001.

Dudik, M., Erhan, D., Langford, J., and Li, L. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4):485–511, 2014.

Garivier, A. and Cappe, O. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceeding of the 24th Annual Conference on Learning Theory*, pp. 359–376, 2011.

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., and Rubin, D. *Bayesian Data Analysis*. Chapman & Hall, 2013.

Hong, J., Kveton, B., Katariya, S., Zaheer, M., and Ghavamzadeh, M. Deep hierarchy in bandits. In *Proceedings of the 39th International Conference on Machine Learning*, 2022a.

Hong, J., Kveton, B., Zaheer, M., and Ghavamzadeh, M. Hierarchical Bayesian bandits. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, 2022b.

Horvitz, D. G. and Thompson, D. J. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685, 1952.

Ionides, E. Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2):295–311, 2008.

Jin, Y., Yang, Z., and Wang, Z. Is pessimism provably efficient for offline RL? In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Kaufmann, E., Cappe, O., and Garivier, A. On Bayesian upper confidence bounds for bandit problems. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, pp. 592–600, 2012.

Kveton, B., Konobeev, M., Zaheer, M., Hsu, C.-W., Mladenov, M., Boutilier, C., and Szepesvari, C. Meta-Thompson sampling. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Lake, B., Salakhutdinov, R., and Tenenbaum, J. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Lam, S. and Herlocker, J. MovieLens Dataset. http://grouplens.org/datasets/movielens/, 2016.

Lattimore, T. and Szepesvari, C. *Bandit Algorithms*. Cambridge University Press, 2019.

Laurent, B. and Massart, P. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302–1338, 2000.

Lazaric, A. and Ghavamzadeh, M. Bayesian multi-task reinforcement learning. In *Proceedings of the 27th International Conference on Machine Learning*, 2010.

Li, G., Ma, C., and Srebro, N. Pessimism for offline linear contextual bandits using ℓ_p confidence sets. In *Advances in Neural Information Processing Systems 35*, 2022.

Li, L., Chu, W., Langford, J., and Schapire, R. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, 2010.

Li, S., Abbasi-Yadkori, Y., Kveton, B., Muthukrishnan, S., Vinay, V., and Wen, Z. Offline evaluation of ranking policies with click models. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1685–1694, 2018.

Lindley, D. and Smith, A. Bayes estimates for the linear model. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(1):1–18, 1972.

Lu, X. and Van Roy, B. Information-theoretic confidence bounds for reinforcement learning. In *Advances in Neural Information Processing Systems 32*, 2019.

Moradipari, A., Turan, B., Abbasi-Yadkori, Y., Alizadeh, M., and Ghavamzadeh, M. Parameter and feature selection in stochastic linear bandits. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Peleg, A., Pearl, N., and Meirr, R. Metalearning linear bandits by prior update. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, 2022.

Robins, J., Rotnitzky, A., and Zhao, L. P. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association*, 89(427):846–866, 1994.

Russo, D. and Van Roy, B. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39 (4):1221–1243, 2014.

Russo, D., Van Roy, B., Kazerouni, A., Osband, I., and Wen, Z. A tutorial on Thompson sampling. *Foundations and Trends in Machine Learning*, 11(1):1–96, 2018.

Simchowitz, M., Tosh, C., Krishnamurthy, A., Hsu, D., Lykouris, T., Dudik, M., and Schapire, R. Bayesian decision-making under misspecified priors with applications to meta-learning. In *Advances in Neural Information Processing Systems 34*, 2021.

Strehl, A., Langford, J., Li, L., and Kakade, S. Learning from logged implicit exploration data. In *Advances in Neural Information Processing Systems 23*, 2010.

Swaminathan, A. and Joachims, T. Counterfactual risk minimization: Learning from logged bandit feedback. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 814–823, 2015.

Swaminathan, A., Krishnamurthy, A., Agarwal, A., Dudik, M., Langford, J., Jose, D., and Zitouni, I. Off-policy evaluation for slate recommendation. In *Advances in Neural Information Processing Systems 30*, 2017.

Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

Wan, R., Ge, L., and Song, R. Metadata-based multi-task bandits with Bayesian hierarchical models. In *Advances in Neural Information Processing Systems 34*, 2021.

Wan, R., Ge, L., and Song, R. Towards scalable and robust structured bandits: A meta-learning framework. *CoRR*, abs/2202.13227, 2022. URL `https://arxiv.org/abs/2202.13227`.

A. Proof of Theorem 1

The proof is under the assumption that the logged dataset $\mathcal{D}$ is fixed and the model parameter θ_* is random. Specifically, since we conduct a Bayesian analysis, we condition on all available observations and have $\theta_* \mid \mathcal{D} \sim \mathcal{N}(\hat{\theta}, \hat{\Sigma})$. We start with the concentration of the model parameter. To simplify notation, we define $r(x, a) = r(x, a; \theta_*)$.

Lemma 3. *Let*

$$E = \{\forall x \in \mathcal{X}, a \in \mathcal{A} : |r(x, a) - \hat{r}(x, a)| \leq c(x, a)\}$$

be the event that all high-probability confidence intervals hold. Then $\mathbb{P}\left(E \,|\, \mathcal{D}\right) \geq 1 - \delta$.

Proof. We start with the Cauchy–Schwarz inequality,

$$r(x, a) - \hat{r}(x, a) = \phi(x, a)(\theta_* - \hat{\theta}) = \phi(x, a)\hat{\Sigma}^{\frac{1}{2}}\hat{\Sigma}^{-\frac{1}{2}}(\theta_* - \hat{\theta}) \leq \|\phi(x, a)\|_{\hat{\Sigma}}\|\theta_* - \hat{\theta}\|_{\hat{\Sigma}^{-1}}\,.$$

To prove our claim, we show that $\|\theta_* - \hat{\theta}\|_{\hat{\Sigma}^{-1}} \leq \sqrt{5d\log(1/\delta)}$ holds conditioned on $\mathcal{D}$ with probability at least $1 - \delta$. The proof uses that $\theta_* - \hat{\theta} \mid \mathcal{D} \sim \mathcal{N}(\mathbf{0}_d, \hat{\Sigma})$. Because of that, $\hat{\Sigma}^{-\frac{1}{2}}(\theta_* - \hat{\theta}) \mid \mathcal{D}$ is a d-dimensional vector of independent standard normal variables. Thus $(\theta_* - \hat{\theta})^\top \hat{\Sigma}^{-1}(\theta_* - \hat{\theta}) \mid \mathcal{D}$ is a chi-squared random variable with d degrees of freedom. Then, by Lemma 1 of Laurent & Massart (2000),

$$\begin{aligned}
\mathbb{P}\left(\|\theta_* - \hat{\theta}\|_{\hat{\Sigma}^{-1}} \geq \alpha \,\middle|\, \mathcal{D}\right) &= \mathbb{P}\left((\theta_* - \hat{\theta})^\top \hat{\Sigma}^{-1}(\theta_* - \hat{\theta}) \geq 5d\log(1/\delta) \,\middle|\, \mathcal{D}\right) \\
&\leq \mathbb{P}\left((\theta_* - \hat{\theta})^\top \hat{\Sigma}^{-1}(\theta_* - \hat{\theta}) \geq 2\sqrt{d\log(1/\delta)} + 2\log(1/\delta) + d \,\middle|\, \mathcal{D}\right) \\
&= \mathbb{P}\left((\theta_* - \hat{\theta})^\top \hat{\Sigma}^{-1}(\theta_* - \hat{\theta}) - d \geq 2\sqrt{d\log(1/\delta)} + 2\log(1/\delta) \,\middle|\, \mathcal{D}\right) \leq \delta\,.
\end{aligned}$$

The first inequality holds under the assumption that $\delta < (0, e^{-1}]$, which implies $\log(1/\delta) \geq 1$ and $\log(1/\delta) \geq \sqrt{\log(1/\delta)}$. This completes our proof. □

We use Lemma 3 to bound the suboptimality of $\hat{\pi}$ in any context by the confidence interval width induced by π_*.

Lemma 4. *Let* $\hat{\pi}(x) = \arg\max_{a \in \mathcal{A}} L(x, a)$. *Then on event* E *(Lemma 3),*

$$r(x, \pi_*(x)) - r(x, \hat{\pi}(x)) \leq 2c(x, \pi_*(x))$$

holds jointly for all contexts $x \in \mathcal{X}$.

Proof. For any context $x \in \mathcal{X}$, we can decompose $r(x, \pi_*(x)) - r(x, \hat{\pi}(x))$ as

$$\begin{aligned}
r(x, \pi_*(x)) - r(x, \hat{\pi}(x)) &= r(x, \pi_*(x)) - L(x, \hat{\pi}(x)) + L(x, \hat{\pi}(x)) - r(x, \hat{\pi}(x)) \\
&\leq [r(x, \pi_*(x)) - L(x, \pi_*(x))] + [L(x, \hat{\pi}(x)) - r(x, \hat{\pi}(x))]\,.
\end{aligned}$$

Now we bound each term separately. On event E, we have $r(x, \pi_*(x)) - \hat{r}(x, \pi_*(x)) \leq c(x, \pi_*(x))$ and thus

$$r(x, \pi_*(x)) - L(x, \pi_*(x)) = r(x, \pi_*(x)) - \hat{r}(x, \pi_*(x)) + c(x, \pi_*(x)) \leq 2c(x, \pi_*(x))\,.$$

Again, on event E, we have $\hat{r}(x, \hat{\pi}(x)) - r(x, \hat{\pi}(x)) \leq c(x, \hat{\pi}(x))$ and thus

$$L(x, \hat{\pi}(x)) - r(x, \hat{\pi}(x)) = \hat{r}(x, \hat{\pi}(x)) - r(x, \hat{\pi}(x)) - c(x, \hat{\pi}(x)) \leq 0\,.$$

Finally, we combine the above two inequalities and get

$$r(x, \pi_*(x)) - r(x, \hat{\pi}(x)) \leq 2c(x, \pi_*(x))\,.$$

This completes the proof. □

In the rest of the analysis, we fix θ_* and the only randomness is due to $X \sim P_\times$. On event E in Lemma 3, we have

$$\begin{aligned} V(\pi_*;\theta_*) - V(\hat{\pi};\theta_*) &= \mathbb{E}\left[r(X,\pi_*(X)) - r(X,\hat{\pi}(X))\right] \leq 2\mathbb{E}\left[c(X,\pi_*(X))\right] \\ &= 2\sqrt{5d\log(1/\delta)}\,\mathbb{E}\left[\sqrt{\phi(X,\pi_*(X))^\top\hat{\Sigma}\phi(X,\pi_*(X))}\right] \\ &\leq 2\sqrt{5d\log(1/\delta)}\sqrt{\mathbb{E}\left[\phi(X,\pi_*(X))^\top\hat{\Sigma}\phi(X,\pi_*(X))\right]}\,. \end{aligned} \tag{15}$$

The first inequality is by Lemma 4. The second inequality follows from the concavity of the square root.

The last step is an upper bound on the expected confidence interval width. Let $\Gamma = \Sigma_0^{-1} + \gamma\sigma^{-2}nG_*$. By Assumption 2, $\hat{\Sigma}^{-1} \succeq \Gamma$ and thus $\hat{\Sigma} \preceq \Gamma^{-1}$. So, for any policy π_*, we have

$$\begin{aligned} \mathbb{E}\left[\phi(X,\pi_*(X))^\top\hat{\Sigma}\phi(X,\pi_*(X))\right] &\leq \mathbb{E}\left[\phi(X,\pi_*(X))^\top\Gamma^{-1}\phi(X,\pi_*(X))\right] \\ &= \mathbb{E}\left[\mathrm{tr}(\Gamma^{-\frac{1}{2}}\phi(X,\pi_*(X))\phi(X,\pi_*(X))^\top\Gamma^{-\frac{1}{2}})\right] \\ &= \mathrm{tr}(\Gamma^{-\frac{1}{2}}G_*\Gamma^{-\frac{1}{2}}) \\ &= \mathrm{tr}(G_*\Gamma^{-1}) = \mathrm{tr}((\Sigma_0^{-1}G_*^{-1} + \gamma\sigma^{-2}nI_d)^{-1}) \\ &\leq \frac{d}{\lambda_d(\Sigma_0^{-1}G_*^{-1} + \gamma\sigma^{-2}nI_d)}\,. \end{aligned}$$

The first inequality follows from Assumption 2. The first equality holds because $v^\top v = \mathrm{tr}(vv^\top)$ for any $v \in \mathbb{R}^d$. The next three equalities use that the expectation of the trace is the trace of the expectation, the cyclic property of the trace, and the definition of matrix inverse. The last inequality follows from $\mathrm{tr}(A^{-1}) \leq d\lambda_1(A^{-1}) = d\lambda_d^{-1}(A)$, which holds for any PSD matrix $A \in \mathbb{R}^{d\times d}$.

Now we apply basic eigenvalue identities and inequalities, and get

$$\begin{aligned} \lambda_d(\Sigma_0^{-1}G_*^{-1} + \gamma\sigma^{-2}nI_d) &= \lambda_d(\Sigma_0^{-1}G_*^{-1}) + \gamma\sigma^{-2}n = \lambda_d((G_*\Sigma_0)^{-1}) + \gamma\sigma^{-2}n = \frac{1}{\lambda_1(G_*\Sigma_0)} + \gamma\sigma^{-2}n \\ &\geq \frac{1}{\lambda_1(G_*)\lambda_1(\Sigma_0)} + \gamma\sigma^{-2}n \geq \frac{1}{\lambda_1(\Sigma_0)} + \gamma\sigma^{-2}n = \lambda_d(\Sigma_0^{-1}) + \gamma\sigma^{-2}n\,. \end{aligned}$$

The last inequality uses $\lambda_1(G_*) \leq 1$, which follows from Assumption 1.

To finalize the proof, we chain all inequalities starting from (15) and get that

$$V(\pi_*;\theta_*) - V(\hat{\pi};\theta_*) \leq \sqrt{5d\log(1/\delta)}\sqrt{\frac{4d}{\lambda_d(\Sigma_0^{-1}) + \gamma\sigma^{-2}n}}$$

holds on event E, which occurs with probability at least $1-\delta$ for $\theta_* \mid \mathcal{D} \sim \mathcal{N}(\hat{\theta},\hat{\Sigma})$. This completes the proof.

B. Frequentist Single-Task Analysis

In this section, we derive a frequentist counterpart to the bound in Theorem 1.

Theorem 5. *Fix dataset $\mathcal{D}$ and let that the rewards be drawn independently as $Y_t - \phi(X_t,A_t)^\top\theta_* \sim \mathrm{SubG}(\sigma^2)$ for some $\sigma > 0$. Let $\hat{\pi}(x) = \arg\max_{a\in\mathcal{A}} L(x,a)$. Then for any $\theta_* \in \Theta$ such that $\|\theta_*\|_2 \leq \kappa$ holds, any γ such that Assumption 2 holds, and any $\delta \in (0,1)$,*

$$V(\pi_*;\theta_*) - V(\hat{\pi};\theta_*) \leq \left(\sqrt{2\log(2\,|\Phi|\,/\delta)} + \kappa\lambda_d^{-\frac{1}{2}}(\Sigma_0)\right)\sqrt{\frac{4d}{\lambda_d(\Sigma_0^{-1}) + \gamma\sigma^{-2}n}}$$

holds with probability at least $1-\delta$, where $\Phi \subseteq \mathbb{R}^d$ is the set of all feature vectors.

The above result compares to the Bayesian error bound in Theorem 1 as follows. Under the assumption that Φ is an ε-grid over $[0,1]^d$, we get $\log(2\,|\Phi|\,/\delta) = O(d\log(1/\varepsilon\delta))$ and the main difference in the bounds is $\kappa\lambda_d^{-\frac{1}{2}}(\Sigma_0)$ in Theorem 5.

To prove Theorem 1, we start with the concentration of the model parameter and define $r(x,a) = r(x,a;\theta_*)$, similarly to Appendix A. We also use $\phi_t = \phi(X_t, A_t)$.

Lemma 6. *Let*

$$E = \{\forall x \in \mathcal{X}, a \in \mathcal{A} : |r(x,a) - \hat{r}(x,a)| \leq c(x,a)\}$$

be the event that all high-probability confidence intervals hold, where

$$c(x,a) = \sqrt{2\log(2\,|\Phi|\,/\delta) + \kappa\lambda_d^{-\frac{1}{2}}(\Sigma_0)}\|\phi(x,a)\|_{\hat{\Sigma}}\,.$$

Then $\mathbb{P}\left(E\right) \geq 1 - \delta$*.*

Proof. We start with the observation that the regularized least-squares estimate of θ_* can be expressed as

$$\begin{aligned}\hat{\theta} &= \sigma^{-2}(\Sigma_0^{-1} + G)^{-1}\sum_{t=1}^n \phi_t Y_t \\ &= \sigma^{-2}(\Sigma_0^{-1} + G)^{-1}\sum_{t=1}^n \phi_t(Y_t - \phi_t^\top\theta_*) + (\Sigma_0^{-1} + G)^{-1}(\Sigma_0^{-1} + G)\theta_* - (\Sigma_0^{-1} + G)^{-1}\Sigma_0^{-1}\theta_* \\ &= \sigma^{-2}(\Sigma_0^{-1} + G)^{-1}\sum_{t=1}^n \phi_t(Y_t - \phi_t^\top\theta_*) + \theta_* - (\Sigma_0^{-1} + G)^{-1}\Sigma_0^{-1}\theta_*\,.\end{aligned}$$

Therefore, for any $\phi \in \Phi$,

$$\phi^\top(\hat{\theta} - \theta_*) = \sigma^{-2}\sum_{t=1}^n \phi^\top(\Sigma_0^{-1} + G)^{-1}\phi_t(Y_t - \phi_t^\top\theta_*) - \phi^\top(\Sigma_0^{-1} + G)^{-1}\Sigma_0^{-1}\theta_*\,. \tag{16}$$

Now note that $\sigma^{-2}\sum_{t=1}^n \phi^\top(\Sigma_0^{-1} + G)^{-1}\phi_t(Y_t - \phi_t^\top\theta_*)$ is a weighted sum of independent sub-Gaussian random variables $Y_t - \phi_t^\top\theta_* \sim \mathrm{SubG}(\sigma^2)$. By definition, its variance proxy is bounded from above as

$$\begin{aligned}\sigma^{-2}\sum_{t=1}^n \phi^\top(\Sigma_0^{-1} + G)^{-1}\phi_t\phi_t^\top(\Sigma_0^{-1} + G)^{-1}\phi &= \phi^\top(\Sigma_0^{-1} + G)^{-1}G(\Sigma_0^{-1} + G)^{-1}\phi \\ &\leq \phi^\top(\Sigma_0^{-1} + G)^{-1}\phi = \|\phi\|_{\hat{\Sigma}}^2\,,\end{aligned}$$

where the last inequality follows from $G \preceq \Sigma_0^{-1} + G$. By the concentration of sub-Gaussian random variables, we have

$$\mathbb{P}\left(\left|\sigma^{-2}\sum_{t=1}^n \phi^\top(\Sigma_0^{-1} + G)^{-1}\phi_t(Y_t - \phi_t^\top\theta_*)\right| \geq \sqrt{2\log(2/\delta)}\|\phi\|_{\hat{\Sigma}}\right) \leq \delta\,.$$

To bound the second term in (16), we apply the Cauchy–Schwarz inequality and get

$$\phi^\top\hat{\Sigma}\Sigma_0^{-1}\theta_* \leq \|\Sigma_0^{-1}\theta_*\|_{\hat{\Sigma}}\|\phi\|_{\hat{\Sigma}} = \sqrt{\theta_*^\top\Sigma_0^{-1}\hat{\Sigma}\Sigma_0^{-1}\theta_*}\|\phi\|_{\hat{\Sigma}} \leq \sqrt{\theta_*^\top\Sigma_0^{-1}\theta_*}\|\phi\|_{\hat{\Sigma}} \leq \kappa\lambda_d^{-\frac{1}{2}}(\Sigma_0)\|\phi\|_{\hat{\Sigma}}\,.$$

The second and third inequalities follow from $\hat{\Sigma} \preceq \Sigma_0$ and $\theta_*^\top\Sigma_0^{-1}\theta_* \leq \kappa^2\lambda_d^{-1}(\Sigma_0)$, respectively. In the next step, we chain all inequalities starting from (16) and get that

$$\mathbb{P}\left(\left|\phi^\top(\hat{\theta} - \theta_*)\right| \geq \left(\sqrt{2\log(2/\delta)} + \kappa\lambda_d^{-\frac{1}{2}}(\Sigma_0)\right)\|\phi\|_{\hat{\Sigma}}\right) \leq \delta$$

holds for any $\phi \in \Phi$ with probability at least $1 - \delta$. To finalize the proof, we apply a union bound over all ϕ. □

The rest of the proof proceeds exactly as in Appendix A, since that proof is for any model parameter θ_* on event E. This completes the proof of Theorem 5.

C. Proof of Theorem 2

The proof is under the assumption that the logged dataset $\mathcal{D}$ is fixed and the task parameter $\theta_{s,*}$ is random. In particular, since we conduct a Bayesian analysis, we condition on all available observations and have $\theta_{s,*} \mid \mathcal{D} \sim \mathcal{N}(\hat{\theta}_s, \hat{\Sigma}_s)$, where $\hat{\theta}_s = \tilde{\Sigma}_s(\Sigma_0^{-1}\bar{\mu} + B_s)$ and $\hat{\Sigma}_s$ are derived in Section 3.2. We start with the concentration of the task parameter. To simplify notation, let $r_s(x, a) = r(x, a; \theta_{s,*})$.

Lemma 7. *Let*

$$E = \{\forall x \in \mathcal{X}, a \in \mathcal{A} : |r_s(x, a) - \hat{r}_s(x, a)| \leq c_s(x, a)\}$$

be the event that all high-probability confidence intervals in task $s \in \mathcal{S}$ hold. Then $\mathbb{P}(E \mid \mathcal{D}) \geq 1 - \delta$.

Proof. The proof is analogous to Lemma 3, since only the mean and covariance of $\theta_{s,*} \mid \mathcal{D}$ changed, and this is reflected in the definitions of $\hat{r}_s(x, a)$ and $c_s(x, a)$. □

On event E in Lemma 7, similarly to Lemma 4, we have that

$$r_s(x, \pi_{s,*}(x)) - r_s(x, \hat{\pi}_s(x)) \leq 2c_s(x, \pi_{s,*}(x))$$

holds for all contexts $x \in \mathcal{X}$ with probability at least $1 - \delta$. Since the above bound holds for any context, we can use use it to bound the suboptimality of $\hat{\pi}_s$ by the expected confidence interval width induced by $\pi_{s,*}$.

In the rest of the analysis, we fix $\theta_{s,*}$ and the only randomness is due to $X \sim P_{\mathsf{x}}$. On event E in Lemma 7, we have

$$\begin{aligned} V(\pi_{s,*}; \theta_{s,*}) - V(\hat{\pi}_s; \theta_{s,*}) &= \mathbb{E}\left[r_s(X, \pi_{s,*}(X)) - r_s(X, \hat{\pi}_s(X))\right] \leq 2\mathbb{E}\left[c_s(X, \pi_{s,*}(X))\right] \\ &= 2\sqrt{5d\log(1/\delta)}\, \mathbb{E}\left[\sqrt{\phi(X, \pi_{s,*}(X))^\top \hat{\Sigma}_s \phi(X, \pi_{s,*}(X))}\right] \\ &\leq 2\sqrt{5d\log(1/\delta)}\sqrt{\mathbb{E}\left[\phi(X, \pi_{s,*}(X))^\top (\tilde{\Sigma}_s \Sigma_0^{-1} \bar{\Sigma} \Sigma_0^{-1} \tilde{\Sigma}_s + \tilde{\Sigma}_s) \phi(X, \pi_{s,*}(X))\right]}\,. \end{aligned} \tag{17}$$

These steps are the same as in (15). The latter term, which represents the conditional task uncertainty, is bounded exactly as in Theorem 1,

$$\mathbb{E}\left[\phi(X, \pi_{s,*}(X))^\top \tilde{\Sigma}_s \phi(X, \pi_{s,*}(X))\right] \leq \frac{d}{\lambda_d(\Sigma_0^{-1}) + \gamma\sigma^{-2} n_s}\,.$$

For the former term, which represents the hyper-parameter uncertainty, we have

$$\begin{aligned} \mathbb{E}\left[\phi(X, \pi_{s,*}(X))^\top \tilde{\Sigma}_s \Sigma_0^{-1} \bar{\Sigma} \Sigma_0^{-1} \tilde{\Sigma}_s \phi(X, \pi_{s,*}(X))\right] &= \mathrm{tr}(G_{s,*} \tilde{\Sigma}_s \Sigma_0^{-1} \bar{\Sigma} \Sigma_0^{-1} \tilde{\Sigma}_s) \\ &\leq d\lambda_1(G_{s,*} \tilde{\Sigma}_s \Sigma_0^{-1} \bar{\Sigma} \Sigma_0^{-1} \tilde{\Sigma}_s)\,. \end{aligned}$$

To bound the maximum eigenvalue, we further proceed as

$$\begin{aligned} \lambda_1(G_{s,*} \tilde{\Sigma}_s \Sigma_0^{-1} \bar{\Sigma} \Sigma_0^{-1} \tilde{\Sigma}_s) &\leq \lambda_1(G_{s,*}) \lambda_1(\tilde{\Sigma}_s \Sigma_0^{-1}) \lambda_1(\bar{\Sigma}) \lambda_1(\Sigma_0^{-1} \tilde{\Sigma}_s) \\ &\leq \lambda_1(\bar{\Sigma}) = \frac{1}{\lambda_d(\Sigma_q^{-1} + \sum_{z \in \mathcal{S}} (\Sigma_0 + G_z^{-1})^{-1})}\,. \end{aligned}$$

The second inequality uses $\lambda_1(G_{s,*}) \leq 1$, which follows from Assumption 1, and $\lambda_1(\tilde{\Sigma}_s \Sigma_0^{-1}) \leq 1$. Finally, we apply basic eigenvalue identities and inequalities, and get

$$\begin{aligned} \lambda_d\left(\Sigma_q^{-1} + \sum_{z \in \mathcal{S}} (\Sigma_0 + G_z^{-1})^{-1}\right) &\geq \lambda_d(\Sigma_q^{-1}) + \sum_{z \in \mathcal{S}} \lambda_d((\Sigma_0 + G_z^{-1})^{-1}) \\ &= \lambda_d(\Sigma_q^{-1}) + \sum_{z \in \mathcal{S}} \lambda_1^{-1}(\Sigma_0 + G_z^{-1}) \\ &\geq \lambda_d(\Sigma_q^{-1}) + \sum_{z \in \mathcal{S}} \frac{1}{\lambda_1(\Sigma_0) + \lambda_1(G_z^{-1})} \\ &\geq \lambda_d(\Sigma_q^{-1}) + \sum_{z \in \mathcal{S}} \frac{1}{\lambda_1(\Sigma_0) + \gamma^{-1}\sigma^2 \lambda_1(G_{z,*}^{-1}) n_z^{-1}}\,, \end{aligned}$$

where we use Assumption 3 in the last inequality.

To finalize the proof, we chain all inequalities starting from (17) and get that

$$V(\pi_{s,*}; \theta_{s,*}) - V(\hat{\pi}_s; \theta_{s,*}) \leq \sqrt{5d \log(1/\delta)} \sqrt{\frac{4d}{\lambda_d(\Sigma_q^{-1}) + \sum_{z \in \mathcal{S}} (\lambda_1(\Sigma_0) + \gamma^{-1} \sigma^2 \lambda_1(G_{z,*}^{-1}) n_z^{-1})}}$$

holds on event E, which occurs with probability at least $1 - \delta$ for $\theta_{s,*} \mid \mathcal{D}$. This completes the proof of the first claim in Theorem 2.

Note that the bound depends on $\lambda_1(G_{z,*}^{-1})$, which can be large when $\lambda_d(G_{z,*})$ is small. We can eliminate this dependence under the additional assumption in Theorem 2. Under that assumption, all matrices are diagonal and thus commute, and

$$\begin{aligned} \lambda_1(G_{s,*} \tilde{\Sigma}_s \Sigma_0^{-1} \bar{\Sigma} \Sigma_0^{-1} \tilde{\Sigma}_s) = \lambda_1(G_{s,*} \bar{\Sigma} \tilde{\Sigma}_s \Sigma_0^{-1} \Sigma_0^{-1} \tilde{\Sigma}_s) \leq \lambda_1(G_{s,*} \bar{\Sigma}) &= \lambda_d^{-1}(\bar{\Sigma}^{-1} G_{s,*}^{-1}) \\ &= \frac{1}{\lambda_d(\Sigma_q^{-1} G_{s,*}^{-1} + \sum_{z \in \mathcal{S}} (G_{s,*} \Sigma_0 + G_{s,*} G_z^{-1})^{-1})} \,. \end{aligned}$$

Finally, we bound the minimum eigenvalue from below using basic eigenvalue identities and inequalities,

$$\begin{aligned} \lambda_d \left(\Sigma_q^{-1} G_{s,*}^{-1} + \sum_{z \in \mathcal{S}} (G_{s,*} \Sigma_0 + G_{s,*} G_z^{-1})^{-1} \right) &\geq \lambda_d(\Sigma_q^{-1}) \lambda_1^{-1}(G_{s,*}) + \sum_{z \in \mathcal{S}} \lambda_1^{-1}(G_{s,*} \Sigma_0 + G_{s,*} G_z^{-1}) \\ &\geq \lambda_d(\Sigma_q^{-1}) + \sum_{z \in \mathcal{S}} \frac{1}{\lambda_1(G_{s,*}) \lambda_1(\Sigma_0) + \lambda_1(G_{s,*} G_z^{-1})} \\ &\geq \lambda_d(\Sigma_q^{-1}) + \sum_{z \in \mathcal{S}} \frac{1}{\lambda_1(\Sigma_0) + \gamma^{-1} \sigma^2 n_z^{-1}} \,. \end{aligned}$$

In the last two inequalities, we use that $\lambda_1(G_{s,*}) \leq 1$. In the last inequality, we also use that Assumption 3 holds for any task parameter, including $\theta_{z,*} = \theta_{s,*}$. Thus $G_z \succeq \gamma \sigma^{-2} n_z G_{s,*}$ and $G_z^{-1} \preceq \gamma^{-1} \sigma^2 n_z^{-1} G_{s,*}^{-1}$. This completes the proof of the second claim in Theorem 2.

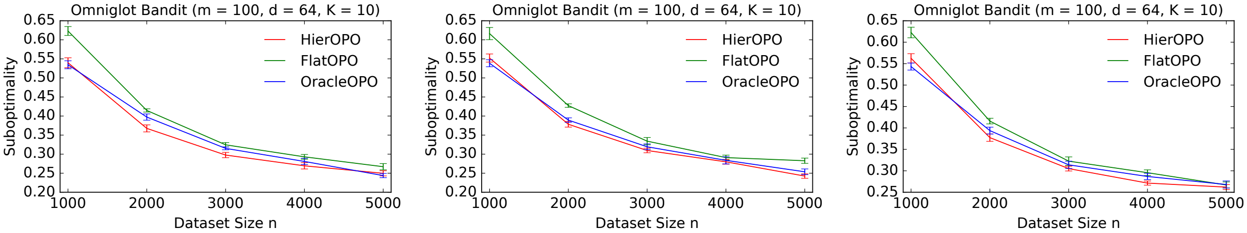

Figure 4. Evaluation of off-policy algorithms on the image classification using Omniglot using three randomly selected alphabets.

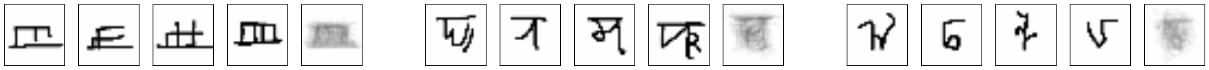

Figure 5. Visualization of the three alphabets used for evaluation. The first four images are randomly selected characters from the alphabet. The fifth is a visualization of the estimation of hyper-parameter μ_* using `HierOPO`, by interpolating the estimated hyper-parameter among characters in the alphabet. Note that the hyper-parameter captures common structures among different characters in the alphabet.

D. Additional Experiment on Image Classification

In this section, we consider an additional experiment based on online image classification using a real-world dataset commonly used in meta-learning. We consider using Omniglot (Lake et al., 2015), which is a dataset of 1623 handwritten characters from 50 different alphabets and contains 20 examples per character. We train a 4-layer CNN to classify the characters from 30 of the alphabets, and use to extract $d = 64$ features using characters from 30 alphabets.

The remaining 20 alphabets are used to evaluate the algorithms as in Section 7. We create a multi-task contextual bandit using the test dataset as follows. First, an alphabet is sampled uniformly at random from a subset of alphabets, which we reserve for evaluation. Then for each task, a character is uniformly chosen from the alphabet as the positive class. By leveraging that all tasks correspond to characters from the same alphabet, we ensure that tasks have some hierarchical relationship. In each round of a particular task, $K = 10$ images from the dataset are chosen, one of which is guaranteed to be of the chosen character. The context is a concatenation of the extracted d-dimensional feature vectors from the CNN trained on the training alphabets for the corresponding chosen images. The reward for selecting an image of the positive class is sampled from $\mathsf{Ber}(0.9)$; otherwise, the reward is sampled from $\mathsf{Ber}(0.1)$.

We estimate the hierarchical model in Section 3.2 using the CNN trained on the training set. We estimate the hyper-prior parameters μ_q and Σ_q using the mean and covariance, respectively, of the features of all the images in the test set. Then for each alphabet, we set μ_* and Σ_0 to be the mean and covariance of the features of images in the alphabet. Recall that μ_*, Σ_0 are unknown to all baselines except `OracleOPO`.

In Figure 4, we report results for three randomly selected alphabets from the test set over 10 random runs, where each run consists of choosing $m = 100$ characters, generating a dataset of size 4n, and running each algorithm on the dataset. Note that here, `HierOPO` and `OracleOPO` both outperform `FlatOPO` initially, but ultimately begin to converge in performance when the dataset size n is large (particularly in the third alphabet). This is because `HierOPO`, `OracleOPO` assume a Gaussian hierarchical structure over characters in an alphabet, which is likely violated. However, as shown in Figure 5, our algorithm `HierOPO` is still able to learn commonalities between characters in an alphabet, such as shape and curvature, which leads to improved performance when n is small.