Offline-to-Online Hyperparameter Transfer for Stochastic Bandits

Dravyansh Sharma[1], **Arun Suggala**[2]
[1]TTIC
[2]Google DeepMind
dravy@ttic.edu, arunss@google.com

Abstract

Classic algorithms for stochastic bandits typically use hyperparameters that govern their critical properties such as the trade-off between exploration and exploitation. Tuning these hyperparameters is a problem of great practical significance. However, this is a challenging problem and in certain cases is information theoretically impossible. To address this challenge, we consider a practically relevant transfer learning setting where one has access to offline data collected from several bandit problems (tasks) coming from an unknown distribution over the tasks. Our aim is to use this offline data to set the hyperparameters for a new task drawn from the unknown distribution. We provide bounds on the inter-task (number of tasks) and intra-task (number of arm pulls for each task) sample complexity for learning near-optimal hyperparameters on unseen tasks drawn from the distribution. Our results apply to several classic algorithms, including tuning the exploration parameters in UCB and LinUCB and the noise parameter in GP-UCB. Our experiments indicate the significance and effectiveness of the transfer of hyperparameters from offline problems in online learning with stochastic bandit feedback.

Introduction

Bandit optimization is a very important framework for sequential decision making with numerous applications, including recommendation systems (Li et al. 2010), healthcare (Tewari and Murphy 2017), AI for Social Good (AI4SG) (Mate et al. 2022), hyperparameter tuning in deep learning (Bergstra et al. 2011). Over the years, numerous works have designed optimal algorithms for bandit optimization in various settings (Garivier and Cappé 2011; Abbasi-Yadkori, Pál, and Szepesvári 2011; Whitehouse, Wu, and Ramdas 2023). One of the key challenges in deploying these algorithms in practice is setting their hyperparameters appropriately. Some examples of these hyperparameters include the confidence width parameter in UCB (Auer, Cesa-Bianchi, and Fischer 2002) and the noise variance parameter σ^2 in GP-UCB (Srinivas et al. 2010). While one could rely on theory-suggested hyperparameters, they often turn out to be too optimistic and lead to suboptimal performance in practice. This is because these hyperparameters are meant to guard the algorithm against worst-case problems and are not necessarily optimal for typical problems arising in a domain. To see this, consider a Multi-Armed Bandit (MAB) instance with two arms, with rewards of each arm drawn from a uniform distribution over an unknown unit width interval. Suppose, the problems encountered in practice are such that the two distributions are well separated without any overlap. Then, the optimal choice of the UCB confidence width parameter is 0 (*i.e.,* there is no need for exploration). This is in contrast to the theory suggested choice of 1, which incurs an exploration penalty. This simplistic example illustrates the importance of choosing hyperparameters appropriately in practice.

Hyperparameter selection, though well-studied in offline learning (Stone 1974; Efron 1992), remains an active research area in bandit optimization. Several recent studies have attempted to address this problem. One particularly popular approach in this line of work is the design of meta-bandit algorithms that treat each hyperparameter as an expert and adaptively select the best expert by running another bandit algorithm on top (also known as *corralling* algorithms) (Agarwal et al. 2017). However, the current algorithms for corralling and their theoretical guarantees are not satisfactory. For instance, consider the stochastic MAB problem, and consider the problem of picking the UCB confidence width parameter. The corralling algorithm of Agarwal et al. (2017) incurs a regret overhead of $O(\sqrt{MT})$ compared to the regret of the best hyperparameter (where M is the size of the hyperparameter search space). This overhead can be quite large when M is large. Furthermore, their algorithm requires the base algorithms to satisfy certain stability conditions. However, UCB is known to satisfy this condition *only* for certain values of the hyperparameter. Recent works of Arora, Marinov, and Mohri (2021) improved the regret guarantees of Agarwal et al. (2017) by considering stochastic bandits. But their regret bounds, when specialized to UCB confidence width tuning, are worse than the regret bounds obtained using the theory-suggested hyperparameter.

Alternative approaches, such as variance-aware algorithms like UCB-V (Mukherjee et al. 2018), hyperparameter-free algorithms (Zimmert and Seldin 2021; Ito 2021), and instance optimal algorithms such as KL-UCB (Garivier and Cappé 2011) still have certain hidden hyperparameters that need careful tuning in practice. For example, UCB-V assumes the reward is bounded between $[0, 1]$. Similarly, the parameter-free algorithms of Zimmert and Seldin (2021); Ito (2021), and instance optimal algorithms assume the reward is bounded

between $[0, 1]$. Such assumptions do not always hold (or are not tight) in practice, leading to suboptimal performance. For example, if the true rewards lie in $[0, 0.5]$ instead of $[0, 1]$, one could choose a smaller exploration parameter in KL-UCB and achieve better regret bounds. In summary, even in the canonical MAB setting, optimal hyperparameter selection for UCB is still an open problem.

In this work, we first ask the following question: *Is it possible to choose good hyperparameters for a given bandit algorithm that perform nearly as well as the best possible hyperparameters, on any given problem instance?*[1] Interestingly, we answer this question negatively, showing that even in the simplest problem of stochastic multi-armed bandits (MABs), determining the best hyperparameter for UCB is information-theoretically impossible.

To address the challenge of hyperparameter selection in bandit algorithms, we propose a data-driven approach. We assume that the learner has access to historical data from *similar* problem instances to the one at hand. This is a reasonable assumption in practical applications of bandit optimization such as *learning-rate* tuning in deep learning, where we often have offline data collected from previous *learning-rate* tuning runs on similar tasks. Our goal is to leverage this side information to find a good hyperparameter that approximately maximizes the expected reward of the bandit algorithm.

We introduce a new framework for hyperparameter tuning of stochastic bandit algorithms, leveraging related historical data as side information. Our approach assumes that problem instances within a specific application domain are sampled from an unknown distribution, and past data from this distribution is available to the learner. Effective hyperparameter tuning in our framework corresponds to minimizing two key quantities: (a) *Inter-task* sample complexity: The number of historical problem instances needed for effective learning, and (b) *Intra-task* sample complexity: The number of arm pulls or data points required from each individual task. A key contribution of our work is the derivation of provable bounds for both inter-task and intra-task sample complexities. These bounds guarantee that the learned hyperparameters are close in performance to the optimal domain-specific choice on unseen tasks drawn from the distribution. Our framework is broadly applicable to several widely-used bandit algorithms, including: (a) tuning the confidence width or exploration parameters in UCB and LinUCB (Abbasi-Yadkori, Pál, and Szepesvári 2011), (b) tuning the noise parameter in GP-UCB (Srinivas et al. 2010). Our experimental results demonstrate the effectiveness and significance of transferring hyperparameter knowledge from offline data to online bandit optimization.

Related Work

This section reviews relevant research on corralling, and transfer learning in bandits. For a review of other topics including model selection in bandits, Bayesian bandits, and data-driven hyperparameter selection in other ML problems, we direct the reader to the Appendix.

[1]A problem instance is defined as a set of arms and their reward distributions.

Corralling. Corralling bandits (Agarwal et al. 2017; Cutkosky, Das, and Purohit 2020; Arora, Marinov, and Mohri 2021; Luo et al. 2022; Ding et al. 2022) is a recent line of work which involves design of bandit algorithms that aim to find the best algorithm from among a finite collection of bandit algorithms. These remarkable techniques have a wide range of applications, including model selection and adapting to misspecification (Foster et al. 2020; Pacchiano et al. 2020). These results also imply an approach for online hyperparameter tuning in bandits, given a finite collection of hyperparameters. However, as discussed in the introduction, these algorithms don't achieve the optimal rates for UCB hyperparameter selection in MAB. Moreover, these approaches are fully online, for which we have impossibility results (see Theorem 1). In contrast, in this work we study bandit hyperparameter selection over continuous parameter domain. Also, we consider a multitask setting where the goal is to learn a good hyperparameter from a collection of offline tasks, and transfer it to the online bandit tasks. Bouneffouf and Claeys (2018) developed a corralling style meta algorithm for hyperparameter selection in contextual bandits Angermueller et al. (2020) provided an algorithm that performs corralling to pick the best bandit algorithm for the design of biological sequences. However, both these works are mostly empirical in nature. Kang, Hsieh, and Lee (2024) consider the continuous parameter domain, but make Lipschitzness assumptions, which are not guaranteed for arbitrary distributions. In fact, even for Bernoulli arm rewards in the MAB setting, the expected rewards can be shown to be a piecewise constant function of the exploration parameter, which is not Lipschitz, and consequently their zooming algorithm based approach cannot be applied.

Transfer Learning. Transfer learning in stochastic bandits, across several related tasks, has received recent attention from the community (Yogatama and Mann 2014). But unlike our work, most of these works have focused on transferring the knowledge of the reward model from one task to another. Our work is complementary to these works as we focus on transferring the knowledge of hyperparameters. Kveton et al. (2020) considered a setting similar to ours, where the bandit instances are sampled from an unknown distribution. The authors studied model learning using gradient descent for simpler policies like explore-then-commit (under nice arm-reward distributions), for which the expected reward as a function of the parameter is concave and easier to optimize. This structure does not hold for several typical reward distributions (e.g. Bernoulli). Our results hold for general unknown reward distributions and more powerful UCB-based algorithm families. Azar, Lazaric, and Brunskill (2013) considered a sequential arrival of tasks, but for the much simpler setting of finitely many models or bandit problems (finite Π in our notation) which are known to the learner. In a similar line of work, Khodak et al. (2023) designed a meta-algorithm to set the initialization, and other hyperparameters of Online Mirror Descent algorithm, based on past tasks. But their work considers transfer learning from one online task to another; in contrast, we consider transfer learning from offline tasks to online tasks. Swersky, Snoek, and Adams (2013); Wang et al.

(2024) showed that learning priors from historic data helped improve the performance of GP-UCB. We note that these works are mostly empirical in nature and do not provide any sample complexity bounds. Apart from bandit feedback, transfer learning of hyperparameters in similar tasks has also been studied in online learning with *full information* feedback (Finn et al. 2019; Khodak, Balcan, and Talwalkar 2019).

Preliminaries

A stochastic online learning problem with bandit feedback consists of a repeated game played over T rounds. In round $t \in [T]$, the player plays an arm $a_t \in [n]$ from a finite set of n arms and the environment simultaneously selects a reward function $r_t : [n] \to \mathbb{R}_{\geq 0}$. In the stochastic setting, the environment draws a reward vector $\mathbf{r}_t$ as an independent sample from some fixed (but unknown) distribution over $\mathbb{R}^n_{\geq 0}$, and the reward is simply $\mathbf{r}_t(a) = \mathbf{r}_{ta}$ for $a \in [n]$. Finally, the player observes and accumulates the reward $r_t(a_t)$ corresponding (only) to the selected arm a_t. This is the well-studied stochastic MAB setting. A standard measure of the player's performance is the pseudo-regret given by

$$R_T = \max_{a \in [n]} \mathbb{E}\left[\sum_{t=1}^{T} r_t(x_t, a) - r_t(x_t, a_t)\right]$$

where the expectation is taken over the randomness of both the player and the environment. The expected average regret is given by $\overline{R_T} := R_T/T$. Let μ_i denote the mean reward of arm $i \in [n]$ and $\Delta_i := \max_j \mu_j - \mu_i$ denote the gap in the mean arm reward relative to the best arm. We will use the shorthand $\mu_{[n]}\{\mu_{l^*} \to \mu'_{l^*}\}$ to denote that the l^*-th entry of the tuple $\mu_{[n]} = (\mu_1, \ldots, \mu_n)$ is updated to μ'_{l^*}.

Impossibility of Hyperparameter Tuning

We now present lower bounds showing that (fully online) optimal hyperparameter selection is not always possible. Before we do that, we first quantify the notion of "optimal" hyperparameter selection. Consider a family of online learning algorithms $\mathcal{A} = \{A_\rho : \rho \in \mathcal{P}\}$, parameterized by hyperparameter ρ. An example of $\mathcal{A}$ is the set of UCB policies, with ρ being the scale parameter multiplied with the confidence width. Let Π be a collection of stochastic multi-armed bandit problems. In hyperparameter tuning, our goal is to design a meta algorithm $\widetilde{A}$ for choosing an appropriate ρ which can compete with the best possible hyperparameter, on any problem instance in Π. To be precise, we want $\widetilde{A}$ to satisfy the following consistency condition for any $P \in \Pi$: $\lim_{T\to\infty} R_T(\widetilde{A}; P)/R_T(A_{\rho^*}; P) = 1$. Here, $\rho^* = \operatorname{argmin}_{\rho \in \mathcal{P}} R_T(A_\rho; P)$ is the best hyperparameter for the problem P. We note that similar notions of consistency have been studied in the context of hyperparameter selection in offline learning (Kearns 1995; Yang 2007). In fact, hyperparameter selection techniques such as cross-validation are known to satisfy such consistency properties. The following result shows that consistency is not possible even in the simplest problem of MAB with rewards sampled from a Gaussian distribution.

Theorem 1. *Let Π be the set of MAB problems with arm rewards sampled from Gaussian distributions with variance belonging to the set $[0, B^2]$. Let $\mathcal{A}$ be the set of UCB policies, with ρ being the scale parameter multiplied with the confidence width. Then for any meta algorithm $\widetilde{A}$, there exists a problem $P \in \Pi$ that satisfies the following bound:* $\lim_{T\to\infty} R_T(\widetilde{A}; P)/R_T(A_{\rho^*}; P) > 1$.

The proof builds on standard distribution-dependent lower bounds (Cappé et al. 2013, see Appendix). This result motivates our framework which uses offline bandit runs on similar problem instances for hyperparameter transfer.

Formal Framework for Transfer Learning

Given a stochastic online learning problem (say multi-armed bandits), let Π denote the set of problems (or tasks) of interest. That is, each $P \in \Pi$ defines an online learning problem. For example, if Π is a collection of stochastic multi-armed bandit problems, then P could correspond to a fixed product distribution of arm rewards. We also fix a (potentially infinite) family of online learning algorithms $\mathcal{A}$, parameterized by a set $\mathcal{P} \subseteq \mathbb{R}^d$ of d real (hyper-)parameters. Let A_ρ denote the algorithm in the family $\mathcal{A}$ parameterized by $\rho \in \mathcal{P}$. For any fixed time horizon T, the performance of any fixed algorithm on any fixed problem is given by some bounded loss metric (e.g. the expected average regret of the algorithm) $l_T : \Pi \times \mathcal{P} \to [0, H]$, i.e. $l_T(P, \rho)$ measures the performance on problem $P \in \Pi$ of algorithm $A_\rho \in \mathcal{A}$. The utility of a fixed algorithm A_ρ from the family is given by $l^\rho_T : \Pi \to [0, H]$, with $l^\rho_T(P) = l_T(P, \rho)$. We will be interested in the structure of the *dual class* of functions $l^P_T : \mathcal{P} \to [0, H]$, with $l^P_T(\rho) = l^\rho_T(P)$, which measure the performance of all algorithms of the family for a fixed problem $P \in \Pi$.

We assume an unknown distribution $\mathcal{D}$ over the problems in Π. We further have a collection of "offline" problems which we can use to learn a good value of the algorithm parameter ρ that works well on average on a random "test" problem drawn from $\mathcal{D}$. We are interested in the sample complexity of the number of "offline" problems that are sufficient to learn a near-optimal ρ over $\mathcal{D}$. Formally, the learner is given a collection $\{P_1, \ldots, P_N\} \sim \mathcal{D}^N$ of *offline* problems for each of which the rewards have been collected according to some policy over time horizon T_o, which we refer to as the *intra-task* complexity. The learner learns a hyperparameter $\hat{\rho}$ based on these offline runs. A *test* problem is given by a random $P \sim \mathcal{D}$ on which the loss metric $l_T(P, \hat{\rho})$ is measured over an online game of T rounds. The (ϵ, δ) *sample complexity* of the learner is the number N of offline problems sufficient to guarantee that learned parameter $\hat{\rho}$ is near-optimal with high probability, i.e. with probability at least $1 - \delta$,

$$\left|\mathbb{E}_{P\sim\mathcal{D}}[l_T(P, \hat{\rho})] - \min_{\rho \in \mathcal{P}} \mathbb{E}_{P\sim\mathcal{D}}[l_T(P, \rho)]\right| \leq \epsilon.$$

Derandomized Dual Complexity

We will define a useful quantity to measure the inherent challenge in learning the best hyperparameter for an unknown problem distribution $\mathcal{D}$. For any offline problem P,

let $\mathbf{z}$ denote the random coins used in drawing the arm rewards according to the corresponding distribution D_P, and $l_T^{P,\mathbf{z}}(\rho)$ denote the corresponding derandomized dual function, i.e. $l_T^P(\rho) = \mathbb{E}_{\mathbf{z}}[l_T^{P,\mathbf{z}}(\rho)]$. Intuitively, we can think of fixing $\mathbf{z}$ as drawing the rewards according to D_P for the entire time horizon T in advance (revealed as usual to the online learner), and taking an expectation over $\mathbf{z}$ gives the expected loss or reward according to P. More concretely, we have $\mathbf{z} = (z_{ti})_{i\in[n],t\in[T]}$ with each z_{ti} drawn i.i.d. from the uniform distribution over $U([0,1])$. If D_i denotes the reward distribution for arm i and F_i its cumulative density function, then the reward $\mathbf{r}_{ti}$ is given by $F_i^{-1}(z_{ti})$.

For typical parameterized stochastic bandit algorithms, we will show that $l_T^{P,\mathbf{z}}(\rho)$ is a piecewise constant function of ρ, i.e. the parameter space $\mathcal{P}$ can be partitioned into finitely many connected regions $\{\mathcal{P}_i\}$ such that $l_T^{P,\mathbf{z}}(\rho) = \sum_i c_i \mathbf{I}[\rho \in \mathcal{P}_i]$ where $c_i \in \mathbb{R}$, $\mathbf{I}[\cdot]$ is the 0-1 valued indicator function, $\bigcup_i \mathcal{P}_i = \mathcal{P}$ and $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$ for $i \neq j$. Let $q(f)$ denote the number of pieces $\{\mathcal{P}_i\}$ over which a piecewise constant function $f : \mathcal{P} \to \mathbb{R}$ is defined. We define the *derandomized dual complexity* of problem distribution $\mathcal{D}$ w.r.t. algorithms parameterized by $\mathcal{P}$ as follows.

Definition 1. *Suppose that the derandomized dual function $l_T^{P,\mathbf{z}}(\rho)$ is a piecewise constant function. The derandomized dual complexity of $\mathcal{D}$ w.r.t. $\mathcal{P}$ is given by $Q_{\mathcal{D}} = \mathbb{E}_{P\sim\mathcal{D}}\mathbb{E}_{\mathbf{z}\sim D_P} q(l_T^{P,\mathbf{z}}(\cdot))$.*

$Q_{\mathcal{D}}$ provides a distribution-dependent complexity measure that will be useful to bound the sample complexity as well as the intratask complexity of learning the best parameter over $\mathcal{D}$. Moreover, it may be empirically estimated over a collection of offline problems sampled from $\mathcal{D}$.

Sample Complexity of Hyperparameter Tuning

We proceed to provide a general *inter-task* sample complexity bound for learning one dimensional parameters, i.e. $\mathcal{P} \subset \mathbb{R}$. We state below our result as a uniform convergence guarantee and provide a proof sketch (full proof in the Appendix).

Theorem 2. *Consider the above setup for any arbitrary $\mathcal{D}$ and suppose the derandomized dual function $l_T^{P,\mathbf{z}}(\rho) : \mathcal{P} \to [0,H]$ is a piecewise constant function. For any $\epsilon, \delta > 0$, N problems $\{P_i\}_{i=1}^N$ sampled from $\mathcal{D}$ with corresponding random coins $\{\mathbf{z}_i\}_{i=1}^N$ such that $N = O\left(\left(\frac{H}{\epsilon}\right)^2 \left(\log Q_{\mathcal{D}} + \log\frac{1}{\delta}\right)\right)$ are sufficient to ensure that with probability at least $1-\delta$, for all $\rho \in \mathcal{P}$, we have that*

$$\left| \frac{1}{N}\sum_{i=1}^{N} l_T^{P_i,\mathbf{z}_i}(\rho) - \mathbb{E}_{P\sim\mathcal{D}} l_T^P(\rho) \right| < \epsilon.$$

Proof Sketch. Fix a problem $P \in \Pi$. Fix the random coins $\mathbf{z}$ used to draw the arm rewards according to D_P for the T rounds. We will use the piecewise loss structure to bound the Rademacher complexity, which would imply uniform convergence guarantees by applying standard learning-theoretic results. Let $\rho, \ldots, \rho_m$ denote a collection of parameter values, with one parameter from each of the $m \leq \sum_{i=1}^N q(l_T^{P_i,\mathbf{z}_i}(\cdot))$ pieces of the dual class functions $l_T^{P_i,\mathbf{z}_i}(\cdot)$ for $i \in [N]$, i.e. across problems in the sample $\{P_1, \ldots, P_N\}$ for some fixed randomizations. Let $\mathcal{F} = \{f_\rho : (P,\mathbf{z}) \mapsto l_T^{P,\mathbf{z}}(\rho) \mid \rho \in \mathcal{P}\}$ be a family of functions on a given sample of instances $S = \{P_i, \mathbf{z}_i\}_{i=1}^N$.

Since the function f_ρ is constant on each of the m pieces, we have the empirical Rademacher complexity, $\hat{R}(\mathcal{F}, S) = \frac{1}{N}\mathbb{E}_\sigma\left[\sup_{j\in[m]} \sum_{i=1}^N \sigma_i v_{ij}\right]$, where $\sigma = (\sigma_1, \ldots, \sigma_m)$ is a tuple of i.i.d. Rademacher random variables, and $v_{ij} := f_{\rho_j}(P_i, \mathbf{z}_i)$. Note that $v^{(j)} := (v_{1j}, \ldots, v_{Nj}) \in [0,H]^N$, and therefore $||v^{(j)}||_2 \leq H\sqrt{N}$, for all $j \in [m]$. An application of Massart's lemma (Massart 2000) now implies $\hat{R}(\mathcal{F}, S) \leq H\sqrt{\frac{2\log\sum_{i=1}^N q(l_T^{P_i,\mathbf{z}_i}(\cdot))}{N}}$.
Taking average over S and applying the Jensen's inequality, gives a bound on the Rademacher complexity

$$R(\mathcal{F}, \mathcal{D}) \leq H\sqrt{\frac{2\log N + 2\log Q_{\mathcal{D}}}{N}}.$$

Classical bounds (Bartlett and Mendelson 2002) now imply the desired sample complexity bound. □

The above result shows that *consistent* hyper-parameter selection is possible as $N \to \infty$, and $\log Q_{\mathcal{D}}$ scales sublinearly in N. In subsequent sections, we derive explicit bounds on the derandomized dual complexity $Q_{\mathcal{D}}$ for several key problems of interest. Theorem 2 not only gives us the inter-task sample complexity, but also reveals an algorithm for finding an ϵ-optimal hyperparameter. In particular, it shows that minimizing $\hat{\rho} := \operatorname{argmin}_{\rho\in\mathcal{P}} \sum_{i=1}^N l_T^{P_i,\mathbf{z}_i}(\rho)$ is enough to guarantee learning a near-optimal parameter. If all arm rewards are observed in the offline data at every time step (called the "full information" setting in online learning), then the intra-task complexity of $T_o = T$ is sufficient to compute $\hat{\rho}$. This is achieved by first estimating $l_T^{P_i,\mathbf{z}_i}(\rho)$ for any ρ, by simulating the bandit algorithm using the observed rewards, and then minimizing the above objective. However, a more realistic assumption is that the offline data was gathered under a bandit learning setting, where only the reward for the pulled arm is observed. In this case, as shown in the following Theorem, having sufficiently long intra-task time horizons T_o for the offline tasks allows us to estimate $l_T^{P_i,\mathbf{z}_i}(\rho)$ for any ρ.

Theorem 3. *There exists an offline data collection policy with $\mathbb{E}[T_o] = \min\{n, Q_{\mathcal{D}}\}T$ and a hyperparameter tuning algorithm that outputs $\hat{\rho}$ which satisfies the following bound with probability at least $1-\delta$,*

$$\mathbb{E}_{P\sim\mathcal{D}}[l_T^P(\hat{\rho})] - \min_\rho \mathbb{E}_{P\sim\mathcal{D}}[l_T^P(\rho)] \leq O\left(\sqrt{\frac{\log Q_{\mathcal{D}} + \log\frac{N}{\delta}}{N}}\right).$$

We now present an offline data collection policy that achieves the above bounds. If the number of arms $n \leq Q_{\mathcal{D}}$, the offline policy is simply to collect the reward for each of the n arms T times. While this is a reasonable policy when n is small, it is not practical when n is large. However, as we show in the sequel, $Q_{\mathcal{D}}$ turns out to be quite small, even when n is large. In this case when $Q_{\mathcal{D}} < n$, the offline policy operates on the

Algorithm 1: UCB(α)

Input: Arms $\{1, \ldots, n\}$, max steps T
Output: Arm pulls $\{A_t \in [n]\}_{t \in [T]}$
for $t = 1, \ldots, n$ **do**
  $A_t \leftarrow t$; /* Pull all arms once. */
  $\hat{\mu}_{A_t} \leftarrow$ observed reward, $t_{A_t} \leftarrow 1$
for $t = n+1, \ldots, T$ **do**
  $A_t \leftarrow \text{argmax}_i \hat{\mu}_i + \sqrt{\frac{\alpha \log t}{t_i}}$; /* $\hat{\mu}_i$ and t_i are the average reward and the number of pulls respectively for arm i */
  Update $\hat{\mu}_{A_t}$ and t_{A_t}

pair $(P, \mathbf{z})$ by sequentially running the algorithm A_ρ for a single ρ value within each interval where the loss function $l_T^{P,\mathbf{z}}(\cdot)$ is constant. The algorithm is restarted with a new ρ value after every T rounds. The hyperparameter tuning algorithm simply minimizes $\min_\rho \sum_{i=1}^N l_T^{P_i,\mathbf{z}_i}(\rho)$. In the sequel, we assume the offline data is collected from these policies. We defer the investigation of finding the optimal offline data collection policy to future work.

Tuning the Exploration Parameter in UCB

The Upper Confidence Bound (UCB) algorithm is a well-known algorithm for the MAB problem. UCB constructs confidence intervals that likely contain the true mean reward for each arm. The core principle is to select the arm with the highest upper confidence bound, embodying the idea of *optimism in the face of uncertainty*. The confidence width parameter α in UCB, which controls the exploration-exploitation trade-off, plays a major role in its performance. We now derive a general bound on the derandomized dual complexity $Q_\mathcal{D}$ for learning α.

Theorem 4. *Let Π be a collection of multi-armed bandit problems with n arms and $\mathcal{D}$ be an arbitrary distribution over Π. Then, for Algorithm 1 parameterized by α, we have* $\log Q_\mathcal{D} = O(n \log T)$.

A proof of the above result appears in the Appendix. We remark that our result makes use of the fact that the arm rewards in the stochastic bandit problem are i.i.d, which implies that the derandomized dual function has polynomially many discontinuities in T (for fixed n). Without this assumption, the number of discontinuities can be as large as $2^{\Omega(T)}$ even for the 2-arm case. Theorems 2 and 3, together with the above result implies an inter-task sample complexity of $\tilde{O}(\frac{n \log T}{\epsilon^2})$ and intra-task complexity of $O(nT)$ to learn ϵ-optimal parameter α. The sample complexity bound can be achieved by the ERM algorithm which solves the following objective: $\min_\alpha \sum_{k=1}^N l_T^{P_k,\mathbf{z}_k}(\alpha)$ (Algorithm 2).

Algorithm 3. An important challenge in implementing ERM is that it involves a minimization over infinitely many α's. We address this by computing the critical points (i.e., points of discontinuities) of the piecewise constant function $l_T^{P_k,\mathbf{z}_k}(\alpha)$. These critical points occur when a slight change to α changes the choice of the arm in UCB (at any time step).

Algorithm 2: TUNEDUCB($\alpha_{\min}, \alpha_{\max}$)

Input: Parameter interval $[\alpha_{\min}, \alpha_{\max}]$, Arm rewards $r_{ijk}, i \in [n], j \in [T], k \in [N]$ from offline data
Output: Learned parameter $\hat{\alpha}$
for *each problem instance* $k \in [N]$ **do**
  for *each arm* $i \in [n]$ **do**
    $t_i \leftarrow 1$
    $R_i[-1] \leftarrow (r_{i2k}, \ldots, r_{iTk})$
  $A_k \leftarrow \alpha$-CRITICALPOINTS($\alpha_{\min}, \alpha_{\max}, (t_1, \ldots, t_n), (r_{11k}, \ldots, r_{n1k}), (R_1[-1], \ldots, R_n[-1])$)
return $\text{argmin}_{\alpha \in \{\alpha_{\min}, \alpha_{\max}\} \cup A_1 \cup \cdots \cup A_N} \sum_{k=1}^N l_T^{P_k,\mathbf{z}_k}(\alpha)$

Algorithm 3 provides an efficient way (runtime proportional to actual number of discontinuities, i.e. expected time complexity is $O(\mathcal{Q}_\mathcal{D})$) to calculate these critical points, making the ERM approach practical. The key idea is to recursively compute the critical points in any interval $[\alpha_l, \alpha_h]$, by locating the first point α_{NEXT} at which an arm different from the best arm for the left end point α_l is selected. In the special case where arm rewards follow a Bernoulli (or categorical) distribution, we can derive a tighter bound for $Q_\mathcal{D}$. A proof of the following Theorem is located in the Appendix.

Theorem 5. *Let Π be a collection of multi-armed bandit problems with n arms and $\mathcal{D}$ be an arbitrary distribution over Π such that the arm rewards take categorical values in $\{0, 1, \ldots, K-1\}$. Then* $\log Q_\mathcal{D} = O(\log KT)$ *for* UCB(α).

LinUCB. A similar analysis can be carried out for the LinUCB algorithm for the stochastic contextual bandits problem (Algorithm 4). In this setting, the learner observes a context vector $(x_{t,i})_{i \in [n]}$ in each round t, assumed to be drawn from a fixed, unknown distribution, and the reward distribution $D_{t,i}$ for the arm i depends on the context $x_{t,i}$. We obtain the following bound on the inter-task sample complexity (proof in the Appendix).

Algorithm 3: α-CRITICALPOINTS($\alpha_l, \alpha_h, t_{[n]}, \mu_{[n]}, R_{[n]}$)

Input: Parameter interval $[\alpha_{\min}, \alpha_{\max}]$, Arm pulls so far t_i, Mean rewards so far μ_i, Future arm rewards $R_i, i \in [n]$.
Output: Learned parameter $\hat{\alpha}$.
if LENGTH(R_i) $= 0$ *for some* $i \in [n]$ **then**
  return $\emptyset$
$l^* \leftarrow \text{argmax}_{i \in [n]} \mu_i + \sqrt{\frac{\alpha_l \log \sum_{j=1}^n t_j}{t_i}}$
$\alpha_{\text{NEXT}} \leftarrow \min_{i \in [n], i \neq l^*} \frac{1}{\sum_{j=1}^n t_j} \left(\frac{\mu_{l^*} - \mu_i}{\frac{1}{\sqrt{t_i}} - \frac{1}{\sqrt{t_{l^*}}}} \right)^2$
$\mu'_{l^*} \leftarrow \frac{\mu_{l^*} t_{l^*} + R_{l^*}[0]}{t_{l^*}+1}$
$\alpha^* \leftarrow \min\{\alpha_h, \alpha_{\text{NEXT}}\}$
$A_1 \leftarrow \alpha$-CRITICALPOINTS($\alpha_l, \alpha^*, t_{[n]}\{t_{l^*} \to t_{l^*}+1\}, \mu_{[n]}\{\mu_{l^*} \to \mu'_{l^*}\}, R_{[n]}\{R_{l^*} \to R_{l^*}[-1]\}$)
if $\alpha_{\text{NEXT}} \geq \alpha_h$ **then**
  return A_1
$A_2 \leftarrow \alpha$-CRITICALPOINTS($\alpha_{\text{NEXT}}, \alpha_h, t_{[n]}, \mu_{[n]}, R_{[n]}$)
return $A_1 \cup \{\alpha_{\text{NEXT}}\} \cup A_2$

Algorithm 4: LINUCB(α)

Input: Arms $\{1, \ldots, n\}$, max steps T, feature dimension d
Output: Arm pulls $\{A_t \in [n]\}_{t \in [T]}$
$K \leftarrow I_d$, $b \leftarrow 0_d$
for $t = 1, \ldots, T$ **do**
  $\theta_t \leftarrow K^{-1} b$
  Observe features $x_{t,1}, \ldots, x_{t,n} \in \mathbb{R}^d$
  $A_t \leftarrow \operatorname{argmax}_i \theta_t^T x_{t,i} + \alpha \sqrt{x_{t,i}^T K^{-1} x_{t,i}}$
  Observe payoff p_{t,A_t}
  Update $K \leftarrow K + x_{t,A_t}^T x_{t,A_t}$ and $b \leftarrow b + x_{t,A_t}^T p_{t,A_t}$

Theorem 6. *Let Π be a collection of contextual bandit problems with n arms and $\mathcal{D}$ be an arbitrary distribution over Π. Then, for* LINUCB(α) *(i.e., Algorithm 4), we have* $\log Q_{\mathcal{D}} = O(T \log n)$.

Our results apply even when the set of arms changes across instances in Π (with n being an upper bound on the number of arms). Suppose now that we have a common set of n arms across all the problems in Π. In this case, in addition to learning the exploration parameter α, we can also learn how to initialize the arm means in the following variant of UCB that incorporates arm priors as hyperparameters.

Learning arm priors. Let $\alpha \in [\alpha_{\min}, \alpha_{\max}]$ denote the exploration parameter and $\hat{\mu}^{\mathbf{0}} = \{\hat{\mu}_1^0, \ldots, \hat{\mu}_n^0\} \in \mathbb{R}^n_{\geq 0}$ be prior (initial) arm means. For any problem $P \in \Pi$ and randomization $\mathbf{z}$, define the dual class functions $l_T^P(\alpha, \hat{\mu}^{\mathbf{0}})$ and $l_T^{P,\mathbf{z}}(\alpha, \hat{\mu}^{\mathbf{0}})$ as above. We have the following sample complexity bound for learning $\alpha, \hat{\mu}^{\mathbf{0}}$ simultaneously.

Theorem 7. *Consider the above setup for any arbitrary $\mathcal{D}$. For any $\epsilon, \delta > 0$, N problems $\{P_i, \mathbf{z}_i\}_{i=1}^N$ sampled from $\mathcal{D}$ with $N = O\left(\left(\frac{H}{\epsilon}\right)^2 \left((n+T)T \log n + \log \frac{1}{\delta}\right)\right)$ are sufficient to ensure that with probability at least $1 - \delta$, for all $\alpha \in [\alpha_{\min}, \alpha_{\max}]$, we have that*

$$\left| \frac{1}{N} \sum_{i=1}^{N} l_T^{P_i, \mathbf{z}_i}(\alpha, \hat{\mu}^{\mathbf{0}}) - \mathbb{E}_{P \sim \mathcal{D}} l_T^P(\alpha, \hat{\mu}^{\mathbf{0}}) \right| < \epsilon.$$

Full proof is located in the Appendix, and employs techniques due to (Bartlett, Indyk, and Wagner 2022).

Tuning the Noise Parameter in GP-UCB

Many problems in reinforcement learning — for example choosing what ads to display to maximize profit in a click-through model (Pandey and Olston 2006), determining the optimal control strategies for a robot (Lizotte et al. 2007), hyperparameter tuning of large machine learning models (Bergstra et al. 2011) — can be formulated as optimizing an unknown noisy function f that is expensive to evaluate. The seminal work of Srinivas et al. (2010) proposed a simple and intuitive Bayesian approach for this problem called the Gaussian Process Upper Confidence Bound (GP-UCB) algorithm and, under implicit smoothness assumptions on f, showed that their algorithm achieves no-regret when optimizing f by a sequence of online evaluations. Their setup formally generalizes the bandit linear optimization problem. A crucial parameter of this algorithm is the noise variance parameter σ^2 which is typically manually tuned. But for many of the above applications, one typically ends up repeatedly solving multiple related problem instances. Learning a value of σ^2 that works well across multiple problem instances is of great practical interest.

Algorithm 5: GP-UCB(σ^2) (Srinivas et al. 2010)

Input: Input space $\mathcal{C}$, GP prior $\mu_0 = 0, \sigma_0$, kernel $k(\cdot, \cdot)$ such that $k(\boldsymbol{x}, \boldsymbol{x}') \leq 1$ for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{C}$, $\{\beta_t\}_{t \in [T]}$.
Output: Point $\{\boldsymbol{x}_t \in \mathcal{C}\}_{t \in [T]}$

for $t = 1, \ldots, T$ **do**
  $\boldsymbol{x}_t \leftarrow \operatorname{argmax}_{\boldsymbol{x} \in \mathcal{C}} \mu_{t-1}(\boldsymbol{x}) + \sqrt{\beta_t} \sigma_{t-1}(\boldsymbol{x})$
  Observe $y_t = f(\boldsymbol{x}_t) + \epsilon_t$, $\epsilon_t \sim N(0, \sigma^2)$
  $\mathbf{k}_t(\boldsymbol{x}) = [k(\boldsymbol{x}_1, \boldsymbol{x}) \ldots k(\boldsymbol{x}_t, \boldsymbol{x})]^T$
  $\mathcal{K}_t = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j \in [t]}$
  Update $\mu_t(\boldsymbol{x}) = \mathbf{k}_t(\boldsymbol{x})^T (\mathcal{K}_t + \sigma^2 I)^{-1} \boldsymbol{y}_t$, where $\boldsymbol{y}_t = [y_1 \ldots y_t]^T$
  Update $\sigma_t(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) - \mathbf{k}_t(\boldsymbol{x})^T (\mathcal{K}_t + \sigma^2 I)^{-1} \mathbf{k}_t(\boldsymbol{x})$

Setup. Consider the problem of maximizing a real-valued function $f : \mathcal{C} \to \mathbb{R}$ over domain $\mathcal{C}$ online. In each round $t = 1, \ldots, T$, the learner selects a point $x_t \in \mathcal{C}$. The learner wants to maximize $\sum_{t=1}^T f(x_t)$, and its performance is measured relative to the best fixed point $x^* = \operatorname{argmax}_{x \in \mathcal{C}} f(x)$. The *instantaneous regret* of the learner is defined as $r_t = f(x^*) - f(x_t)$ and the cumulative regret as $R_T = \sum_{t=1}^T r_t$.

The parameter σ^2 in this algorithm is the noise variance, which is unknown to the learner but needed by the algorithm. An upper bound on the true noise σ^2 is sufficient for the algorithm to work, but a loose upper bound can weaken the regret guarantees of Srinivas et al. (2010). In practice, the parameter is set heuristically for each problem. We consider a set-up similar to the multi-armed bandits problem above. Consider the parameterized family of GP-UCB algorithms given by Algorithm 5 with parameter $\sigma^2 = s \in [s_{\min}, s_{\max}]$ for some $0 < s_{\min} < s_{\max} < \infty$. Let $\mathcal{D}$ be a distribution over some problems, i.e. a distribution over noisy (random) real-valued functions $f : \mathcal{C} \to [0, H]$ with $\mathcal{C} \subset \mathbb{R}^d$. It is typical to discretize the domain $\mathcal{C}$ when computing the argmax in Algorithm 5, and usually $f(\cdot)$ is more expensive to evaluate than the UCB acquisition function $a_t(\boldsymbol{x}) := \mu_t(\boldsymbol{x}) + \sqrt{\beta_t} \sigma_t(\boldsymbol{x})$ for any point $\boldsymbol{x}$ on the finite discretization $\tilde{\mathcal{C}}$ of $\mathcal{C}$ with $|\tilde{\mathcal{C}}| = n$. The following Theorem derives the inter-task sample complexity for learning s.

Theorem 8. *Consider the above setup for any arbitrary $\mathcal{D}$. Let $n = |\tilde{\mathcal{C}}|$. For any $\epsilon, \delta > 0$, N problems $\{P_i, \mathbf{z}_i\}_{i=1}^N$ sampled from $\mathcal{D}$ with $N = O\left(\left(\frac{H}{\epsilon}\right)^2 \left(T \log nT + \log \frac{1}{\delta}\right)\right)$ are sufficient to ensure that with probability at least $1 - \delta$, for all $s \in [s_{\min}, s_{\max}]$, we have that*

$$\left| \frac{1}{N} \sum_{i=1}^{N} l_T^{P_i, \mathbf{z}_i}(s) - \mathbb{E}_{P \sim \mathcal{D}} l_T^P(s) \right| < \epsilon.$$

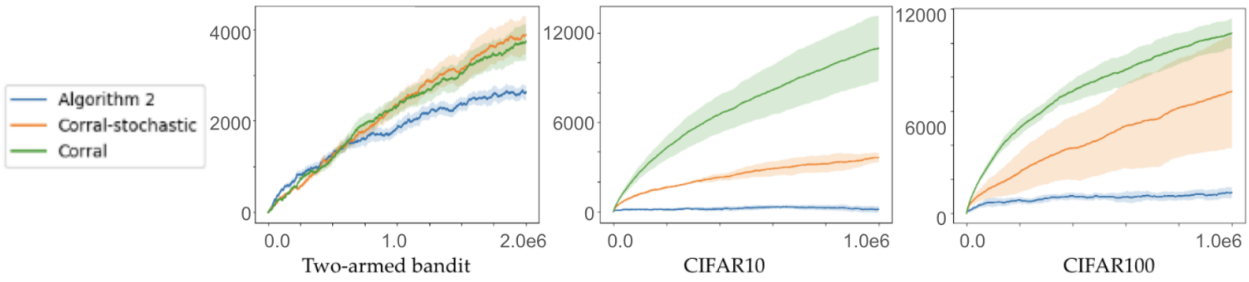

Figure 1: Comparison of Algorithm 2 to corralling based algorithms CORRAL (Agarwal et al. 2017) and CORRAL-STOCHASTIC (Arora, Marinov, and Mohri 2021).

We find that the sample complexity required to accurately learn s increases linearly with T. But this does not pose a significant limitation in applications such as hyperparameter optimization in deep learning, where T is often very small.

Experiments

In this section, we provide empirical evidence for the significance of our hyperparameter transfer framework on real and synthetic data. As baselines, we consider corralling-based algorithms which are quite popular for learning bandit hyperparameters. As described previously, these approaches work by constructing a (finite) band of bandit algorithms corresponding to a grid of hyperparameter values, and running a meta-algorithm for selecting the hyperparameter in each round. The original CORRAL algorithm (Agarwal et al. 2017) uses an OMD (Online Mirror Descent) meta-algorithm with Log-Barrier regularizer, and the CORRAL-STOCHASTIC algorithm (Arora, Marinov, and Mohri 2021) uses a Tsallis-INF regularizer and achieves stronger instance-dependent regret guarantees for stochastic bandits. By exploiting offline data, we out-perform both these corralling-based approaches on real datasets involving tuning the learning rate of neural networks on benchmark datasets.

Synthetic two-armed bandits. We consider a simple two-armed bandits problem with Bernoulli arm rewards (see Appendix for more experiments). Arm 1 draws a reward of 0 or 1 with probability 0.5 each in all tasks. Arm 2 draws a reward of value 1 with probability $0.5+\epsilon$ with $\epsilon \sim \mathcal{N}(0.01, \sigma_b^2 = 0.01)$ and 0 otherwise. Given the small arm gap, this is a challenging problem that needs a lot of exploration.

Hyperparameter tuning for Deep Learning. We also consider the task of tuning the learning rate for training neural networks on image classification tasks. The arms consist of 11 different learning rates $(0.001$,0.002,0.004,0.006,0.008,0.01,0.05,$0.1, 0.2, 0.4, 0.8)$ and the arm reward is given by the classification accuracy of feedforward neural networks trained via SGD (stochastic gradient descent) with that learning rate and a batch size of 64 for 20 epochs. We present our results for CIFAR-10 and CIFAR-100 (Krizhevsky 2009) benchmark image classification datasets. The task distribution is defined by a uniform distribution over the label noise proportions $(0.0, 0.1, 0.2, 0.3)$, and the network depth $(2, 4, 6, 8, 10)$. All our experiments on CIFAR are run on 1 Nvidia A100 GPU.

Setup and Discussion. For each dataset we run Algorithm 2 over $N = 200$ training/offline tasks with time horizon $T_o = 20$, and run corralling for a grid of ten hyperparameter values $\alpha = \{0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100\}$. Figure 1 compares the effectiveness of running UCB with the learned hyperparameter $\hat{\alpha}$ vs. corralling over a grid of hyperparameters over 1000000 time steps (mean regret and standard deviation over 5 iterations). Our algorithm which exploits offline data significantly outperforms corralling based algorithms on real datasets. The key advantage of our approach is that by learning a good hyperparameter offline, we can save significantly on exploration. We verify this hypothesis by considering a synthetic two-armed bandits problem with Bernoulli parameters 0.5 and roughly 0.51, where a large amount of exploration is unavoidable. Even on this challenging synthetic dataset where the arm rewards are extremely close and more exploration is needed, our algorithm beats corralling over a longer time horizon of 2000000 steps. Further details and addtitional experiments showing dependence of regret on α and the number of training tasks N are in the Appendix, where we also empirically estimate $\mathcal{Q}_{\mathcal{D}}$ and show that typical values on natural problems are quite small, implying that our proposed algorithms are sample and computationally efficient in practice. Recall that our inter-task, intra-task and computational complexities scale as $O(\log \mathcal{Q}_{\mathcal{D}})$, $O(\mathcal{Q}_{\mathcal{D}})$, $O(\mathcal{Q}_{\mathcal{D}})$ respectively.

Conclusion, Limitations and Future Work

We study the problem of tuning hyperparameters of stochastic bandit algorithms, given access to offline data. Our setting is motivated by large information theoretic gaps for identifying the best hyperparameter in a fully online fashion, in the bandit setting. We provide a formal framework where the tasks are drawn iid from some distribution and the learner has access to some offline (training) tasks. For tuning the exploration parameter in UCB and the noise parameter in GP-UCB, we obtain bounds on the time horizon and number of the offline tasks needed to obtain any desired generalization performance on the unseen test tasks from the same distribution.

We believe the intra-task sample complexity bounds provided in our work can be improved with more careful arguments. An important question that our work doesn't yet address is: *how to strategically collect offline data to minimize the intra-task sample complexity?* Another direction is to tune hyperparameters beyond UCB-style algorithms.

References

Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved algorithms for linear stochastic bandits. *Advances in Neural Information Processing Systems*, 24.

Agarwal, A.; Luo, H.; Neyshabur, B.; and Schapire, R. E. 2017. Corralling a band of bandit algorithms. In *Conference on Learning Theory*, 12–38. PMLR.

Angermueller, C.; Belanger, D.; Gane, A.; Mariet, Z.; Dohan, D.; Murphy, K.; Colwell, L.; and Sculley, D. 2020. Population-based black-box optimization for biological sequence design. In *International Conference on Machine Learning*, 324–334. PMLR.

Arora, R.; Marinov, T. V.; and Mohri, M. 2021. Corralling stochastic bandit algorithms. In *International Conference on Artificial Intelligence and Statistics*, 2116–2124. PMLR.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2): 235–256.

Azar, M. G.; Lazaric, A.; and Brunskill, E. 2013. Sequential Transfer in Multi-armed Bandit with Finite Set of Models. In *Advances in Neural Information Processing Systems*.

Bartlett, P.; Indyk, P.; and Wagner, T. 2022. Generalization Bounds for Data-Driven Numerical Linear Algebra. In *Conference on Learning Theory*, 2013–2040. PMLR.

Bartlett, P. L.; and Mendelson, S. 2002. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov): 463–482.

Bergstra, J.; Bardenet, R.; Bengio, Y.; and Kégl, B. 2011. Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems*, 24.

Bouneffouf, D.; and Claeys, E. 2018. Hyper-parameter tuning for the contextual bandit. *Association for the Advancement of Artificial Intelligence (AAAI)*.

Cappé, O.; Garivier, A.; Maillard, O.-A.; Munos, R.; and Stoltz, G. 2013. Kullback-Leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 1516–1541.

Cutkosky, A.; Das, A.; and Purohit, M. 2020. Upper confidence bounds for combining stochastic bandits. *arXiv preprint arXiv:2012.13115*.

Ding, Q.; Kang, Y.; Liu, Y.-W.; Lee, T. C. M.; Hsieh, C.-J.; and Sharpnack, J. 2022. Syndicated bandits: A framework for auto tuning hyper-parameters in contextual bandit algorithms. *Advances in Neural Information Processing Systems*, 35: 1170–1181.

Efron, B. 1992. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics: Methodology and distribution*, 569–593. Springer.

Finn, C.; Rajeswaran, A.; Kakade, S.; and Levine, S. 2019. Online meta-learning. In *International Conference on Machine Learning*, 1920–1930. PMLR.

Foster, D. J.; Gentile, C.; Mohri, M.; and Zimmert, J. 2020. Adapting to misspecification in contextual bandits. *Advances in Neural Information Processing Systems*, 33: 11478–11489.

Garivier, A.; and Cappé, O. 2011. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual conference on learning theory*, 359–376. JMLR Workshop and Conference Proceedings.

Ito, S. 2021. Parameter-free multi-armed bandit algorithms with hybrid data-dependent regret bounds. In *Conference on Learning Theory*, 2552–2583. PMLR.

Kang, Y.; Hsieh, C.-J.; and Lee, T. 2024. Online Continuous Hyperparameter Optimization for Generalized Linear Contextual Bandits. *Transactions on Machine Learning Research*.

Kearns, M. 1995. A bound on the error of cross validation using the approximation and estimation rates, with consequences for the training-test split. *Advances in Neural Information Processing Systems*, 8.

Khodak, M.; Balcan, M.-F.; and Talwalkar, A. S. 2019. Adaptive gradient-based meta-learning methods. *Advances in Neural Information Processing Systems*, 32.

Khodak, M.; Osadchiy, I.; Harris, K.; Balcan, M.-F.; Levy, K. Y.; Meir, R.; and Wu, Z. S. 2023. Meta-Learning Adversarial Bandit Algorithms. *Advances in Neural Information Processing Systems*.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. Technical report.

Kveton, B.; Mladenov, M.; Hsu, C.-W.; Zaheer, M.; Szepesvari, C.; and Boutilier, C. 2020. Meta-learning bandit policies by gradient ascent. *arXiv preprint arXiv:2006.05094*.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, 661–670.

Lizotte, D. J.; Wang, T.; Bowling, M. H.; Schuurmans, D.; et al. 2007. Automatic Gait Optimization With Gaussian Process Regression. In *IJCAI*, volume 7, 944–949.

Luo, H.; Zhang, M.; Zhao, P.; and Zhou, Z.-H. 2022. Corralling a larger band of bandits: A case study on switching regret for linear bandits. In *Conference on Learning Theory*, 3635–3684. PMLR.

Massart, P. 2000. Some applications of concentration inequalities to statistics. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, volume 9, 245–303.

Mate, A.; Madaan, L.; Taneja, A.; Madhiwalla, N.; Verma, S.; Singh, G.; Hegde, A.; Varakantham, P.; and Tambe, M. 2022. Field study in deploying restless multi-armed bandits: Assisting non-profits in improving maternal and child health. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 12017–12025.

Mukherjee, S.; Naveen, K.; Sudarsanam, N.; and Ravindran, B. 2018. Efficient-UCBV: An almost optimal algorithm using variance estimates. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Pacchiano, A.; Phan, M.; Abbasi Yadkori, Y.; Rao, A.; Zimmert, J.; Lattimore, T.; and Szepesvari, C. 2020. Model selection in contextual stochastic bandit problems. *Advances in Neural Information Processing Systems*, 33: 10328–10337.

Pandey, S.; and Olston, C. 2006. Handling advertisements of unknown quality in search advertising. *Advances in Neural Information Processing Systems*, 19.

Srinivas, N.; Krause, A.; Kakade, S.; and Seeger, M. W. 2010. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *International Conference on Machine Learning*.

Stone, M. 1974. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society: Series B (Methodological)*, 36(2): 111–133.

Swersky, K.; Snoek, J.; and Adams, R. P. 2013. Multi-task bayesian optimization. *Advances in Neural Information Processing Systems*, 26.

Tewari, A.; and Murphy, S. A. 2017. From ads to interventions: Contextual bandits in mobile health. *Mobile health: sensors, analytic methods, and applications*, 495–517.

Wang, Z.; Dahl, G. E.; Swersky, K.; Lee, C.; Nado, Z.; Gilmer, J.; Snoek, J.; and Ghahramani, Z. 2024. Pre-trained Gaussian processes for Bayesian optimization. *Journal of Machine Learning Research*, 25(212): 1–83.

Whitehouse, J.; Wu, Z. S.; and Ramdas, A. 2023. Improved Self-Normalized Concentration in Hilbert Spaces: Sublinear Regret for GP-UCB. *arXiv preprint arXiv:2307.07539*.

Yang, Y. 2007. Consistency of Cross Validation for Comparing Regression Procedures. *The Annals of Statistics*, 2450–2473.

Yogatama, D.; and Mann, G. 2014. Efficient transfer learning method for automatic hyperparameter tuning. In *Artificial Intelligence and Statistics*, 1077–1085. PMLR.

Zimmert, J.; and Seldin, Y. 2021. Tsallis-inf: An optimal algorithm for stochastic and adversarial bandits. *Journal of Machine Learning Research*, 22(28): 1–49.