Proceedings of Machine Learning Research vol 65:1–27, 2017

Corralling a Band of Bandit Algorithms

Alekh Agarwal ALEKHA@MICROSOFT.COM
Microsoft Research, New York

Haipeng Luo HAIPENG@MICROSOFT.COM
Microsoft Research, New York

Behnam Neyshabur BNEYSHABUR@TTIC.EDU
Toyota Technological Institute at Chicago

Robert E. Schapire SCHAPIRE@MICROSOFT.COM
Microsoft Research, New York

Abstract

We study the problem of combining multiple bandit algorithms (that is, online learning algorithms with partial feedback) with the goal of creating a master algorithm that performs almost as well as the best base algorithm *if it were to be run on its own*. The main challenge is that when run with a master, base algorithms unavoidably receive much less feedback and it is thus critical that the master not starve a base algorithm that might perform uncompetitively initially but would eventually outperform others if given enough feedback. We address this difficulty by devising a version of Online Mirror Descent with a special mirror map together with a sophisticated learning rate scheme. We show that this approach manages to achieve a more delicate balance between exploiting and exploring base algorithms than previous works yielding superior regret bounds.

Our results are applicable to many settings, such as multi-armed bandits, contextual bandits, and convex bandits. As examples, we present two main applications. The first is to create an algorithm that enjoys worst-case robustness while at the same time performing much better when the environment is relatively easy. The second is to create an algorithm that works simultaneously under different assumptions of the environment, such as different priors or different loss structures.

Keywords: bandits, ensemble, adaptive algorithms

1. Introduction

We study the problem of combining suggestions from a collection of online learning algorithms in the partial feedback setting, with the goal of achieving good performance as long as one of these base algorithms performs well for the problem at hand.

For example, suppose a company wants to do personalized advertising using some *contextual bandit* (Langford and Zhang, 2008) algorithm. Different algorithms in the literature outperform others under different environments (e.g. i.i.d or adversarial), making it hard to commit to one of them beforehand. Instead of trying them all once and committing to the best—an inefficient and nonadaptive approach—is it possible to come up with an adaptive and automatic master algorithm whose performance is always competitive with the best of these base algorithms in any environment?

In the full-information setting where the losses for all actions are revealed at each round, this problem can be solved simply by running, for instance, the weighted majority algorithm (Littlestone and Warmuth, 1989). However, this does not directly work in the *bandit* setting, since the base

algorithms whose suggestions were ignored cannot update their internal state. A natural impulse in this case is to run a multi-armed bandit algorithm (such as EXP3 (Auer et al., 2002b)) as the master, treating the base algorithms as arms. By the regret guarantee of a multi-armed bandit algorithm, which states that on average the performance of the master is almost as good as the best arm, it seems that our scenario is perfectly addressed.

However, this reasoning is flawed as the base algorithms are not static arms. While the master algorithm does compete with the base algorithms in terms of their *actual* performance during the run, this performance could be significantly worse than if the base algorithm were run on its own, updating its state after every prediction. For instance, a base algorithm which is exploratory initially but excels later on might quickly fall out of favor with the master, effectively meaning that it never gets to explore enough and reach its good performance regime. Therefore, the real objective of creating such an ensemble is to make sure that the master performs almost as well as the best base algorithm if *it were to be run on its own*. As we will see in this paper, this modified objective leads to an even more delicate explore-exploit trade-off than standard bandit problems.

The most related previous work is by Maillard and Munos (2011) (see also the survey of Bubeck and Cesa-Bianchi (2012, Chapter 4.2)) who studied special cases of our framework. They essentially run EXP4 (Auer et al., 2002b) as the master, with some additional uniform exploration. If the base algorithms are EXP3 or its variants which have $\mathcal{O}(\sqrt{T})$ regret bounds when run by themselves, where T is the number of rounds, these works show $\mathcal{O}(T^{2/3})$ regret for the master—the loss in rates due to the additional uniform exploration. Whether the regret can be improved to $\mathcal{O}(\sqrt{T})$ in this case was left as a major open problem. Feige et al. (2014) model base algorithms as stateful policies and consider a much harder objective that only admits $\Theta(T/\text{poly}(\ln T))$ results.

In this work we present a generic result for this problem in a much more general framework, which includes multi-armed, contextual, and convex bandits, and more (see Section 2), and affirmatively addresses the setting of the open problem in particular. In Section 3, we first show that in general no master can have non-trivial regret even when one of the base algorithms has constant regret, which motivates us to make some very natural stability assumption on the base algorithms. With this assumption, we propose a novel master algorithm, called CORRAL, which manages to explore more actively but adaptively, and achieve similar regret bounds as the best base algorithm as shown in Section 4.

Our solution is based on a special instance of the well-studied Online Mirror Descent framework (see for example (Shalev-Shwartz, 2011)), with a mirror map that in some sense admits the highest possible amount of exploration while keeping the optimal regret. This mirror map was recently studied in (Foster et al., 2016) for a very different purpose of obtaining first-order regret bounds and our analysis is also different. Another key ingredient of our solution is a sophisticated schedule for tuning the learning rates of the master algorithm, which increases the learning rate corresponding to a specific base algorithm when it has relatively low probability of getting feedback. This tuning schedule was also recently used in (Bubeck et al., 2016) in a completely different context of designing computationally efficient convex bandit method.

To show the power of our new approach, in Section 5 we present two scenarios where one can directly use our master algorithm to create a more adaptive solution. The first is to create an algorithm that guarantees strong robustness in the worst case but at the same time can perform much better when the environment is relatively easy (for example, when the data is i.i.d. from a distribution). The second is to create an algorithm that works simultaneously under different models (for example, different priors or different loss structures) and is able to select the correct

model automatically. Besides algorithms from (Bubeck and Slivkins, 2012; Seldin and Slivkins, 2014; Auer and Chiang, 2016) for both stochastic and adversarial multi-armed bandits, our general results are the first of these kinds to the best of our knowledge.[1]

We present several examples of these applications in different settings. For example, going back to contextual bandits example, we have the following result:

Theorem 1 (informal) *There is an efficient contextual bandit algorithm with regret $\widetilde{\mathcal{O}}(\sqrt{T})$ if both contexts and losses are i.i.d, and simultaneously with regret $\widetilde{\mathcal{O}}(T^{3/4})$ if the losses are adversarially chosen, but the contexts are still i.i.d.*

2. Formal Setup

We consider a general online optimization problem with bandit feedback, which can be seen as a repeated game between the environment and the learner. On each round $t = 1, \dots, T$:

1. the environment first reveals some *side information* $x_t \in \mathcal{X}$ to the learner;
2. the learner makes a *decision* $\theta_t \in \Theta$ for some decision space Θ, while simultaneously the environment decides a *loss function* $f_t : \Theta \times \mathcal{X} \to [0, 1]$;
3. finally, the learner incurs and observes (only) the loss $f_t(\theta_t, x_t)$.

For simplicity, we measure the performance of an algorithm by its *(pseudo-)regret*, defined as

$$\sup_{\theta \in \Theta} \mathbb{E}\left[\sum_{t=1}^{T} f_t(\theta_t, x_t) - f_t(\theta, x_t)\right]$$

where the expectation is taken over the randomness of both the player and the environment.[2]

Throughout the paper we will talk about different environments. Formally, an environment $\mathcal{E}$ is a randomized mapping from the history $(x_s, \theta_s, f_s)_{s=1,\dots,t-1}$ to a new outcome (x_t, f_t). Equivalently, one can also assume that all randomness of the environment is drawn ahead of time, which allows us to capture Bayesian settings too. We use the notation $(x_t, f_t) = \mathcal{E}(\theta_1, \dots, \theta_{t-1})$ to make the dependence on the learner's decisions explicit.

This general setup subsumes many bandit problems studied in the literature including multi-armed, convex, and contextual bandits. At a high-level, the decision sets correspond to policies or action sets available to the player, and environments capture assumptions on the adversary such as being oblivious or stochastic. We present an example that instantiates all these quantities concretely at the end of this section, with more detailed examples in Appendix A.

We assume that we are given a set of M bandit algorithms, denoted by $\mathcal{B}_1, \dots, \mathcal{B}_M$, each designed for the general setup above for some environments and decision space $\Theta_i \subset \Theta$. We refer to these as *base algorithms*. We aim to develop a *master algorithm* which makes a decision on each round after receiving suggestions from the base algorithms. We restrict the master to pick amongst the suggestions of the base algorithms so that it does not need to know any details of the base algorithms or the problem itself.

1. Our regret bounds are always at least $\sqrt{T}$ and do not recover results in (Bubeck and Slivkins, 2012; Seldin and Slivkins, 2014; Auer and Chiang, 2016) though.
2. For conciseness, in the rest of the paper we simply call this the regret.

Our goal is to ensure that the performance of the master is not far away from the best base algorithm had it been run separately. As discussed earlier, this is challenging since each base algorithm has access to a much smaller amount of data when run with a master than on its own; nevertheless, we want to compete with the counterfactual in which a single base algorithm drives all of the decisions and receives feedback on every round. We capture the behavior of a base algorithm $\mathcal{B}_i$ using its promised regret bound when run in isolation. Suppose for some (randomized) environment, $\mathcal{B}_i$ produces a sequence $\theta_1^i, \ldots, \theta_T^i \in \Theta_i$, such that the following bound holds:[3]

$$\sup_{\theta \in \Theta_i} \mathbb{E}\left[\sum_{t=1}^{T} f_t(\theta_t^i, x_t) - f_t(\theta, x_t)\right] \leq \mathcal{R}_i(T),$$

for some regret bound $\mathcal{R}_i : \mathbb{N}_+ \to \mathbb{R}_+$. Then, ideally, we might hope that under the same environment, if we run the master with all these base algorithms to make the decisions $\theta_1, \ldots, \theta_T$, we have

$$\sup_{\theta \in \Theta_i} \mathbb{E}\left[\sum_{t=1}^{T} f_t(\theta_t, x_t) - f_t(\theta, x_t)\right] \leq \mathcal{O}(\text{poly}(M)\mathcal{R}_i(T)), \tag{1}$$

where the expectation is taken over the randomness of the master, the base algorithms and the environment. In words, we want the expected loss of the master to be competitive with the expected loss of the best decision θ in the decision space of each base algorithm $\mathcal{B}_i$, up to a level which depends on the regret of $\mathcal{B}_i$. This problem is beguilingly subtle. And in this ideal, aspirational form, there is reason to doubt that such a master algorithm can exist at all in general. Nevertheless, in this paper, we make significant progress toward developing such an algorithm. But to be clear, the results are subject to important caveats and conditions that we state precisely in Section 3.

Example 1 (Contextual bandits) In contextual bandits (Langford and Zhang, 2008), the side information x_t is typically called a context, the decision space Θ is a set of policies $\theta : X \to [K]$ and the loss function takes the form $f_t(\theta, x) = \langle \boldsymbol{c}_t, \boldsymbol{e}_{\theta(x)} \rangle$ for some $\boldsymbol{c}_t \in [0,1]^K$ specifying the loss of each action at round t. (Here and throughout the paper, $[n]$ denotes the set $\{1, \ldots, n\}$, and $\boldsymbol{e}_i$ denotes the i-th standard basis vector.) This problem has been studied under three main environments:

Stochastic contexts and losses: Here the environment is characterized by a fixed distribution from which contexts x_t and losses $\boldsymbol{c}_t$ are drawn i.i.d. The Epoch-Greedy algorithm of Langford and Zhang (2008) suffers an expected regret of $\widetilde{\mathcal{O}}(T^{2/3})$ in this setting, while Agarwal et al. (2014) get the optimal $\widetilde{\mathcal{O}}(\sqrt{T})$ regret at a higher computational cost.

Adversarial contexts or losses: Several authors (Auer, 2002; Chu et al., 2011; Filippi et al., 2010) have studied environments where the contexts are chosen by an adversary, but the losses come from a fixed conditional distribution given x_t. Other authors (Syrgkanis et al., 2016; Rakhlin and Sridharan, 2016) have considered contexts drawn i.i.d. from a fixed distribution, but the losses picked in an adversarial manner. Syrgkanis et al. (2016) have proposed a computationally efficient algorithm which suffers an expected regret of at most $\widetilde{\mathcal{O}}(T^{2/3})$ in this setting.

Adversarial contexts and losses: The EXP4 algorithm of Auer et al. (2002b) incurs an expected regret at most $\widetilde{\mathcal{O}}(\sqrt{T})$ in this most general setting, but is computationally inefficient.

3. In general, regret should also depend on other parameters such as the size of the decision space Θ, but we will treat these parameters as fixed constants and see the regret as solely a function of T when losses are in $[0,1]$.

3. Assumption and Algorithm

Intuitively, the task of the master algorithm appears quite similar to a standard multi-armed bandit problem, with each base algorithm as an arm. However, as hinted in the introduction, this is not the case — the problem admits no non-trivial results without further assumptions. In this vein, we now present a lower bound, and an assumption to avoid it. After understanding the failure of typical algorithms despite making the assumption, we then present our approach.

3.1. Hardness in the Worst Case and A Natural Assumption

We begin with the following hardness result (see Appendix B for the proof).

Theorem 2 *There is an environment and a pair of base algorithms $\mathcal{B}_1, \mathcal{B}_2$ such that either $\mathcal{R}_1(T)$ or $\mathcal{R}_2(T)$ is a constant (independent of T), but for any master algorithm combining $\mathcal{B}_1, \mathcal{B}_2$, the expected regret of the master is at least $\Omega(T)$ (thus, the bound (1) does not hold).*

This lower bound and its proof highlight the main challenge of our problem. The assumption of a good regret bound on each base algorithm when run in isolation in an environment is not sufficient since we can come up with pathological examples where the behavior of the algorithm completely changes when run under a master. Therefore, we next consider natural modifications of the environment of a base algorithm, to which we expect robustness.

Recall that in our setting, the master only observes the loss for the decision suggested by one of the base algorithms it picked (randomly). It is thus natural to create importance-weighted losses for each base algorithm.

To this end, for an environment $\mathcal{E}$, we define the *environment $\mathcal{E}'$ induced by importance weighting*, which is the environment that results when importance weighting is applied to the losses provided by environment $\mathcal{E}$. More precisely, $\mathcal{E}'$ is defined as follows. On each round $t = 1, \ldots, T$,

1. $\mathcal{E}'$ picks an arbitrary sampling probability $p_t \in [0, 1]$ and obtains $(x_t, f_t) = \mathcal{E}(\theta'_1, \ldots, \theta'_{t-1})$.
2. $\mathcal{E}'$ reveals x_t to the learner and the learner makes a decision θ_t.
3. With probability p_t, define $f'_t(\theta, x) = f_t(\theta, x)/p_t$ and $\theta'_t = \theta_t$; with probability $1 - p_t$, define $f'_t(\theta, x) \equiv 0$ and $\theta'_t \in \Theta$ to be arbitrary.
4. $\mathcal{E}'$ reveals the loss $f'_t(\theta_t, x_t)$ to the learner, and passes θ'_t to $\mathcal{E}$.

Such an induced environment is exactly the one that the base algorithms face when run with a master using importance-weighted losses. If the base algorithms have similar performance under $\mathcal{E}$ and $\mathcal{E}'$, then we can exclude the pathological examples which govern our lower bound. However, while the original loss f_t is in $[0, 1]$, the estimated loss f'_t, although an unbiased estimate of f_t, takes values in the larger range $[0, 1/p_t]$, meaning that the range of losses has changed significantly. We therefore define the following notion of stability of the base algorithms, which captures how much an algorithm's regret degrades as a result of the range expanding in this fashion:

Definition 3 *For some $\alpha \in (0, 1]$ and non-decreasing function $\mathcal{R} : \mathbb{N}_+ \to \mathbb{R}_+$, an algorithm with decision space $\Theta_0 \subset \Theta$ is called $(\alpha, \mathcal{R})$-stable with respect to an environment $\mathcal{E}$ if its regret under $\mathcal{E}$ is $\mathcal{R}(T)$, and its regret under any environment $\mathcal{E}'$ induced by importance weighting is*

$$\sup_{\theta \in \Theta_0} \mathbb{E}\left[\sum_{t=1}^{T} f'_t(\theta_t, x_t) - f'_t(\theta, x_t)\right] \leq \mathbb{E}\left[\rho^\alpha\right] \mathcal{R}(T) \tag{2}$$

where $\rho = \max_{t\in[T]} 1/p_t$ (with p_t as in the definition of $\mathcal{E}'$ above), and all expectations are taken over the randomness of both $\mathcal{E}'$ and the algorithm.

This stability assumption intuitively posits that the regret of the algorithm grows at most linearly in the scale of the losses it receives. In the adversarial construction of Theorem 2 (in Appendix B), we can see that this is certainly not the case there. However, for most "reasonable" base algorithms, a linear scaling with $\alpha = 1$ is trivially achievable simply by rescaling the losses. Moreover, note that the second moment of the loss estimate f'_t is also bounded by ρ (instead of ρ^2): $\mathbb{E}_{p_t}[f'_t(\theta_t, x_t)^2] \le \rho$, and as we will see in the sequel, the regret of many natural bandit algorithms does scale as some function of the second moment of the loss sequence. In such cases, it is typical to obtain an exponent α strictly smaller than 1.

There are two seemingly strong parts about this condition. First, the bound requires adaptation to the quantity ρ which is unknown to the algorithm ahead of time. However, this can be easily resolved by a standard doubling trick (Cesa-Bianchi et al., 1997). Second, if an algorithm is designed for an i.i.d. environment $\mathcal{E}$, then we might not expect to have any regret guarantee under $\mathcal{E}'$ since it is not an i.i.d. environment anymore. However, even in this case, due to the special structure of $\mathcal{E}'$, one can still prove stability for many i.i.d. algorithms as we will show later.

In conclusion, our stability condition is a natural and mild requirement for an algorithm. In Appendix D, we show how this condition is satisfied for most existing bandit algorithms, either as is, or by extremely simple modifications (also see Table 1 for a summary).

Armed with the assumption, it is natural to revisit a question from before: can we use any existing multi-armed bandit algorithm as a master and hope to get guarantee (1) under this assumption? It turns out that the answer is still no if we were to use an arbitrary multi-armed bandit algorithm as a master. To see why, consider the classic multi-armed bandit algorithm EXP3 (Auer et al., 2002b) as the master. EXP3 induces probabilities that are exponentially small in the cumulative loss of $\mathcal{B}_i$, meaning that the scaling ρ can grow exponentially large with T. We can mitigate this problem partially by adding additional uniform exploration to EXP3, but one can verify that such modifications unavoidably lead to a major deterioration in the regret (for example $\mathcal{O}(T^{2/3})$ regret of the master even when all the base algorithms have $\mathcal{O}(\sqrt{T})$ regret). This is exactly the issue noted in prior works (Maillard and Munos, 2011; Bubeck and Cesa-Bianchi, 2012), as mentioned in the introduction.

In the next subsection, we present a specific multi-armed bandit algorithm that does address all these issues successfully, and provide results on its performance in the sections that follow.

3.2. Our Algorithm

It is well known that EXP3 belongs to a large family of algorithms called *Online Mirror Descent*, and it is thus natural to ask whether there is a different instance of Online Mirror Descent that solves our problem. Specifically, let $\boldsymbol{p}_t$ be the distribution for picking a base algorithm on round t, and $\boldsymbol{\ell}_t$ be some loss estimator of the base algorithms. Online Mirror Descent updates $\boldsymbol{p}_t$ as follows:

$$\begin{aligned} \nabla\psi_t(\tilde{\boldsymbol{p}}_{t+1}) &= \nabla\psi_t(\boldsymbol{p}_t) - \boldsymbol{\ell}_t \\ \boldsymbol{p}_{t+1} &= \operatorname*{argmin}_{\boldsymbol{p}\in\Delta_M} D_{\psi_t}(\boldsymbol{p}, \tilde{\boldsymbol{p}}_{t+1}) \end{aligned}$$

where $\psi_1, \dots, \psi_T$ are the *mirror maps* and $D_\psi(\boldsymbol{p}, \boldsymbol{q}) = \psi(\boldsymbol{p}) - \psi(\boldsymbol{q}) - \langle \nabla\psi(\boldsymbol{q}), \boldsymbol{p} - \boldsymbol{q}\rangle$ is the *Bregman divergence* associated with ψ. For example, EXP3 with a learning rate η uses negative

Algorithm 1: CORRAL

Input: learning rate η and M base algorithms $\mathcal{B}_1, \dots, \mathcal{B}_M$

Initialize: $\gamma = 1/T$, $\beta = e^{\frac{1}{\ln T}}$, $\eta_{1,i} = \eta$, $\rho_{1,i} = 2M$ for all $i \in [M]$, $\boldsymbol{p}_1 = \bar{\boldsymbol{p}}_1 = \frac{\mathbf{1}}{M}$

Initialize all base algorithms

for $t = 1$ **to** T **do**

  Observe side information x_t, send x_t to $\mathcal{B}_i$ and receive decision θ_t^i for each $i \in [M]$

  Sample $i_t \sim \bar{\boldsymbol{p}}_t$, predict $\theta_t = \theta_t^{i_t}$, observe loss $f_t(\theta_t, x_t)$

  Send $f_t^i(\theta_t^i, x_t)$ to $\mathcal{B}_i$ as feedback for each $i \in [M]$ where $f_t^i(\theta, x) = \frac{f_t(\theta,x)}{\bar{p}_{t,i_t}} \mathbf{1}\{i = i_t\}$

  Update $\boldsymbol{p}_{t+1} = \text{LOG-BARRIER-OMD}(\boldsymbol{p}_t, \frac{f_t(\theta_t,x_t)}{\bar{p}_{t,i_t}} \boldsymbol{e}_{i_t}, \boldsymbol{\eta}_t)$

  Set $\bar{\boldsymbol{p}}_{t+1} = (1-\gamma)\boldsymbol{p}_{t+1} + \gamma \frac{\mathbf{1}}{M}$

  for $i = 1$ **to** M **do**

    if $\frac{1}{\bar{p}_{t+1,i}} > \rho_{t,i}$ **then** set $\rho_{t+1,i} = \frac{2}{\bar{p}_{t+1,i}}$, $\eta_{t+1,i} = \beta \eta_{t,i}$

    else set $\rho_{t+1,i} = \rho_{t,i}$, $\eta_{t+1,i} = \eta_{t,i}$

Algorithm 2: LOG-BARRIER-OMD$(\boldsymbol{p}_t, \boldsymbol{\ell}_t, \boldsymbol{\eta}_t)$

Input: previous distribution $\boldsymbol{p}_t$, current loss $\boldsymbol{\ell}_t$ and learning rate vector $\boldsymbol{\eta}_t$

Output: updated distribution $\boldsymbol{p}_{t+1}$

Find $\lambda \in [\min_i \ell_{t,i},\ \max_i \ell_{t,i}]$ such that $\sum_{i=1}^M \frac{1}{\frac{1}{p_{t,i}} + \eta_{t,i}(\ell_{t,i} - \lambda)} = 1$

Return $\boldsymbol{p}_{t+1}$ such that $\frac{1}{p_{t+1,i}} = \frac{1}{p_{t,i}} + \eta_{t,i}(\ell_{t,i} - \lambda)$

entropy $\psi_t(\boldsymbol{p}) = \frac{1}{\eta} \sum_{i=1}^M p_i \ln p_i$ as the mirror map and updates $p_{t+1,i} \propto \exp(-\eta \sum_{s=1}^t \ell_{s,i})$. As discussed before, due to the exponential weighting, $p_{t,i}$ can be very small and thus more likely lead to the starvation of the base algorithms that perform poorly initially. Various other mirror maps have been proposed in the literature and might be considered for our purpose, most notably the negative Tsallis entropy (Audibert and Bubeck, 2010; Abernethy et al., 2015).

We would suggest, however, that the most suitable mirror map is $\psi_t(\boldsymbol{p}) = -\frac{1}{\eta} \sum_{i=1}^M \ln p_i$, originally suggested in a recent work of Foster et al. (2016). This choice is motivated by our previous discussion which suggests that a good master algorithm should ensure that $1/p_{t,i}$ does not grow too rapidly. Typically, this quantity grows exponentially in $\eta \sum_{s=1}^t \ell_{s,i}$ for EXP3, and polynomially for Tsallis entropy with a degree greater than 1. On the other hand, the mirror map suggested above gives the simple update $p_{t+1,i} = (\eta \sum_{s=1}^t \ell_{s,i} + Z)^{-1}$ where Z is a normalization factor, which means that $1/p_{t,i}$ grows just linearly in $\eta \sum_{s=1}^t \ell_{s,i}$ and appears to be the least extreme weighting amongst all known algorithms that yield $\sqrt{T}$ regret in the bandit setting. Thus, this mirror map is likely to provide a high level of exploration without hurting the regret guarantees. This special instance of Online Mirror Descent is a key component of our algorithm. Since this mirror map resembles the log barrier for the positive orthant (Nesterov and Nemirovskii, 1994), we call it LOG-BARRIER-OMD.[4] Note that while LOG-BARRIER-OMD was proposed in (Foster et al., 2016) to provide a so-called "small-loss" regret bound, here we are not using this special regret bound but a rather different property of the algorithm.

4. Note that this is different from directly using a barrier *for the simplex* as proposed in (Abernethy et al., 2012; Rakhlin and Sridharan, 2013).

OMD methods typically use a decaying learning rate schedule to ensure convergence. Intuitively, this is at odds with our learning goal. If a base algorithm $\mathcal{B}_i$ starts to do well after underperforming for a while, we want to exploit this good performance. We achieve this by instead considering *non-decreasing* learning rate schemes. However, a schedule merely based on the round t is not sufficient since we would like to adjust the learning rate for $\mathcal{B}_i$ based on its performance. Hence, we allow each $\mathcal{B}_i$ to have its own learning rate $\eta_{t,i}$, which corresponds to using $\psi_t(\boldsymbol{p}) = -\sum_{i=1}^{M} \frac{\ln p_i}{\eta_{t,i}}$ as the actual mirror map. The precise setting of $\eta_{t,i}$ is discussed further below.

Our main algorithm, called CORRAL, is presented in Algorithm 1. It is clear that Lines 5, 6 and 7 are essentially performing LOG-BARRIER-OMD over the base algorithms with the usual unbiased loss estimator $\boldsymbol{\ell}_t = \frac{f_t(\theta_t, x_t)}{\bar{p}_{t,i_t}} \boldsymbol{e}_{i_t}$. Note that we sample the base algorithms according to $\bar{\boldsymbol{p}}_t$, a smoothed version of $\boldsymbol{p}_t$ (see Line 8, where $\mathbf{1}$ denotes the all-ones vector), which can be seen as adding another hard constraint to prevent the weighting from becoming too extreme. Moreover, one can verify that LOG-BARRIER-OMD admits a simple update formula as presented in Algorithm 2. Indeed, plugging the gradients of the log barrier mirror map gives $\frac{1}{\tilde{p}_{t+1,i}} = \frac{1}{p_{t,i}} + \eta_{t,i}\ell_{t,i}$. On the other hand, the Bregman divergence is $D_{\psi_t}(\boldsymbol{p}, \boldsymbol{q}) = \sum_{i=1}^{M} \frac{1}{\eta_{t,i}} h\left(\frac{p_i}{q_i}\right)$ with $h(y) = y - 1 - \ln y$, and by standard analysis the projection step can be solved as in Algorithm 2 by finding the Lagrange multiplier λ via a line search.

Line 10 gives our setting of the learning rates $\eta_{t,i}$. This setting is motivated by our analysis, where we show that one of the terms for the regret of CORRAL against $\mathcal{B}_i$ scales roughly as $\sum_{t=1}^{T} \frac{1}{\bar{p}_{t+1,i}} \left(\frac{1}{\eta_{t+1,i}} - \frac{1}{\eta_{t,i}}\right)$. Since $\eta_{t,i}$ is non-decreasing in t, this term is always nonpositive and its magnitude is large when $\bar{p}_{t+1,i}$ is small. This conveys the intuition that when CORRAL puts low probability on $\mathcal{B}_i$, it is incurring negative regret to $\mathcal{B}_i$. While several settings of $\eta_{t,i}$ are plausible, we find it convenient to analyze the evolution of this term if $\eta_{t,i}$ is updated as in Line 10 where it is increased by a factor of β whenever $1/\bar{p}_{t+1,i}$ is larger than some threshold. On each such event, we know that $1/\bar{p}_{t+1,i}$ is large so that we gain a large negative term in the regret. At the same time, $\eta_{T,i}$ is at most a constant factor larger than $\eta_{1,i}$ which is crucial in other parts of our analysis. For ease of bookkeeping, we also maintain the threshold $\rho_{t,i}$ which is always an upper bound on $1/\bar{p}_{t,i}$ (and hence the loss that $\mathcal{B}_i$ receives on round t), and update it in a doubling manner. These thresholds essentially correspond to the quantity ρ in the stability condition. We note that similar non-decreasing learning rate schedule were originally proposed in (Bubeck et al., 2016) for a different problem.

Remark *An alternative for the feedback to the base algorithms and the loss estimator is to let*

$$f_t^i(\theta, x) = \frac{f_t(\theta, x)}{\sum_{j:\theta_t^j = \theta_t^{i_t}} \bar{p}_{t,j}} \mathbf{1}\{\theta_t^i = \theta_t^{i_t}\} \quad \textit{and} \quad \ell_{t,i} = f_t^i(\theta_t^i, x_t)$$

which is less wasteful when Θ is a small set and it is more likely that multiple base algorithms make the same decision. One can verify that this alternative provides the same guarantee as Algorithm 1.

4. Main Results

In this section, we present our main results which give guarantees on the performance of CORRAL when base algorithms satisfy the stability condition defined in Definition 3. We first give a general result before elaborating on its various implications.

Theorem 4 *For any $i \in [M]$, if base algorithm $\mathcal{B}_i$ (with decision space Θ_i) is $(\alpha_i, \mathcal{R}_i)$-stable (recall Defn. 3) with respect to an environment $\mathcal{E}$, then under the same environment* CORRAL *satisfies*

$$\sup_{\theta \in \Theta_i} \mathbb{E}\left[\sum_{t=1}^{T} f_t(\theta_t, x_t) - f_t(\theta, x_t)\right] = \widetilde{\mathcal{O}}\left(\frac{M}{\eta} + T\eta - \frac{\mathbb{E}[\rho_{T,i}]}{\eta} + \mathbb{E}[\rho_{T,i}^{\alpha_i}]\mathcal{R}_i(T)\right), \tag{3}$$

where all expectations are taken over the randomness of Algorithm 1, the base algorithms, and the environment.

In our setup, the master algorithm CORRAL ends up playing the role of the importance-weighting adversary in Definition 3 and the theorem assumes stability of the base algorithms to such modifications of the loss. The regret bound given in Eq. (3) can be interpreted intuitively as follows. The first two terms arise as the typical regret incurred in a standard adversarial multi-armed bandit problem. The next two terms capture the distinct aspects of our hierarchical setup. The last term comes directly from the regret of $\mathcal{B}_i$ relative to Θ_i according to the stability condition. The negative term arises as discussed in Section 3.2. CORRAL induces low probabilities, and hence large $\rho_{T,i}$ on $\mathcal{B}_i$ if it finds that $\mathcal{B}_i$ is being consistently outperformed by some other base algorithm. In this case, the master gets a negative regret with respect to $\mathcal{B}_i$ by increasing the learning rate for $\mathcal{B}_i$. Note that this term scales with $\rho_{T,i}$, which is crucial in obtaining better regret than prior works.

In order to better understand this general result, we now further simplify the theorem for two special cases. We start with the case where the master wants to guarantee low regret against a base algorithm $\mathcal{B}_i$ with $\alpha_i = 1$. That is, the regret of the base algorithm scales linearly with the range of the losses. In this setting, we obtain the following result.

Theorem 5 *Under the conditions of Theorem 4, if $\alpha_i = 1$, then with $\eta = \min\left\{\frac{1}{40\mathcal{R}_i(T)\ln T}, \sqrt{\frac{M}{T}}\right\}$* CORRAL *satisfies:* $\sup_{\theta \in \Theta_i} \mathbb{E}\left[\sum_{t=1}^{T} f_t(\theta_t, x_t) - f_t(\theta, x_t)\right] = \widetilde{\mathcal{O}}\left(\sqrt{MT} + M\mathcal{R}_i(T)\right)$.

Theorem 5 is evidently yielding the ideal bound (1), but with an important caveat. The initial learning rate η needs to be set based on the regret bound $\mathcal{R}_i(T)$ of the base algorithm $\mathcal{B}_i$ that we wish to compete with. One way of interpreting this theorem is the following. Suppose for a given environment, S is the set of all base algorithms which we are interested in competing with and which satisfy the condition with $\alpha_i = 1$. Let $\mathcal{R}_{\max}(T) = \max_{i \in S} \mathcal{R}_i(T)$. Then we get:

Corollary 6 *Under the conditions of Theorem 5, with $\eta = \min\left\{\frac{1}{40\mathcal{R}_{\max}(T)\ln T}, \sqrt{\frac{M}{T}}\right\}$,* CORRAL *satisfies:* $\sup_{\theta \in \Theta_i} \mathbb{E}\left[\sum_{t=1}^{T} f_t(\theta_t, x_t) - f_t(\theta, x_t)\right] = \widetilde{\mathcal{O}}\left(\sqrt{MT} + M\mathcal{R}_{\max}(T)\right)$, *for all $i \in S$.*

The main application of this result is the following situation that resembles model selection problems. One has a collection of base algorithms such that each of them works well under a different environment or decision space, and we want one single robust algorithm that works well simultaneously across all these environments and decision spaces (see Section 5.2 for examples).

However, there is also another related class of applications for which Theorem 5 turns out not to be so helpful. For instance, in contextual bandits, we might consider using for base algorithms: (1) the algorithm of Agarwal et al. (2014), which is nearly statistically optimal but computationally

Algorithm	$\mathcal{R}(T)$	α	Environment
ILOVETOCONBANDITS	$\widetilde{\mathcal{O}}(\sqrt{KT\ln\lvert\Theta\rvert})$	$1/2$	stochastic contextual bandit
BISTRO+	$\mathcal{O}((KT)^{\frac{2}{3}}(\ln\lvert\Theta\rvert)^{\frac{1}{3}})$	$1/3$	hybrid contextual bandit
Epoch-Greedy	$\widetilde{\mathcal{O}}(T^{\frac{2}{3}}\sqrt{K\ln\lvert\Theta\rvert})$	$1/3$	stochastic contextual bandit
EXP4	$\mathcal{O}(\sqrt{KT\ln\lvert\Theta\rvert})$	$1/2$	adversarial contextual bandit
SCRiBLe	$\widetilde{\mathcal{O}}(d^{\frac{3}{2}}\sqrt{T})$	$1/2$	adversarial linear bandit
BGD	$\mathcal{O}(d\sqrt{L}T^{\frac{3}{4}})$	$1/4$	adversarial convex bandit
Thompson Sampling	$\mathcal{O}(\sqrt{TKH(\theta^*)})$	$1/2$	stochastic multi-armed bandit

Table 1: Examples of base algorithms (see Section 5 and Appendix D for details)

somewhat expensive, using a fairly small policy class; and (2) the Epoch-Greedy algorithm of Langford and Zhang (2008), which is statistically suboptimal but computationally cheap, using a larger policy class. If we apply Theorem 5 with these base algorithms, either we suffer a substantially suboptimal regret of $\widetilde{\mathcal{O}}(T^{2/3})$ against both policy classes, or we do not end up with any non-trivial guarantee against the richer class used by Epoch-Greedy. Our next theorem partially alleviates such concerns by exploiting the case when $\alpha_i < 1$ (in fact all our examples admit $\alpha_i < 1$).

Theorem 7 *Under the conditions of Theorem 4, if $\alpha_i < 1$ then* CORRAL *satisfies*

$$\sup_{\theta\in\Theta_i}\mathbb{E}\left[\sum_{t=1}^{T} f_t(\theta_t,x_t)-f_t(\theta,x_t)\right]=\widetilde{\mathcal{O}}\left(\frac{M}{\eta}+T\eta+\mathcal{R}_i(T)^{\frac{1}{1-\alpha_i}}\eta^{\frac{\alpha_i}{1-\alpha_i}}\right).$$

To see why Theorem 7 is useful, consider again the contextual bandit scenario discussed above and assume $\alpha_i = 1/3$ for Epoch-Greedy (we will indeed prove this in Appendix D). In this case, choosing $\eta = \Theta(\frac{1}{\sqrt{T}})$ ensures $\mathcal{O}(\sqrt{T})$ regret against the small policy class while slightly larger $\mathcal{O}(T^{3/4})$ regret against the richer policy class — a guarantee that neither of the base algorithms provides by itself.

5. Applications

We present several concrete examples of using CORRAL in this section. The base algorithms we consider for different problems are listed in Table 1 along with their stability parameters under a specific class of environments. All details and proofs can be found in Appendix D.

5.1. Exploiting Easy Environments with Robust Guarantee

We present two examples below to show how to use CORRAL to create an algorithm that enjoys some robustness guarantee while being able to exploit easy environments and perform much better than in the worst case.

5.1.1. CONTEXTUAL BANDITS

We consider the setting of Example 1. It is in general difficult to derive efficient contextual bandit algorithms without any assumptions on the set Θ and the environment. Prior works usually

assume access to an offline ERM oracle that, given a set of training examples $(x_s, \boldsymbol{c}_s)_{s=1,\dots,t}$, outputs the policy that minimizes the loss on this training set. We consider three such algorithms: ILOVETOCONBANDITS (Agarwal et al., 2014), BISTRO+ (Syrgkanis et al., 2016) and the simplest, explore-first version of Epoch-Greedy (Langford and Zhang, 2008), denoted by $\mathcal{B}_1, \mathcal{B}_2$ and $\mathcal{B}_3$ respectively. In addition, we also consider a classic but inefficient algorithm EXP4 (Auer et al., 2002b), denoted by $\mathcal{B}_4$. All these base algorithms satisfy the stability condition but under different environments, as stated in Lemmas 16, 17, 18, and 19.

Now assuming the context distribution is known, we first combine $\mathcal{B}_1$, which exploits the case when losses are also stochastic (that is, drawn from a conditional distribution $\mathcal{D}(\cdot|x)$), and $\mathcal{B}_2$, which provides a safe guarantee even when the losses are generated adversarially. The following result is a direct application of Theorems 5 and 7.

Corollary 8 *Suppose we run* CORRAL *with two base algorithms:* ILOVETOCONBANDITS *and* BISTRO+ *with learning rate* $\eta = 1/\sqrt{KT \ln |\Theta|}$ *and* $\Theta = \Theta_1 = \Theta_2$. *Assuming* $x_1, \dots, x_T$ *are generated independently from a fixed and known distribution, we have:*

1. $\sup_{\theta \in \Theta} \mathbb{E}\left[\sum_{t=1}^T f_t(\theta_t, x_t) - f_t(\theta, x_t)\right] = \widetilde{\mathcal{O}}\left((KT)^{\frac{3}{4}}\sqrt{\ln |\Theta|}\right)$ *for adversarial costs;*
2. $\sup_{\theta \in \Theta} \mathbb{E}\left[\sum_{t=1}^T f_t(\theta_t, x_t) - f_t(\theta, x_t)\right] = \widetilde{\mathcal{O}}(\sqrt{KT \ln |\Theta|})$, *if* $\boldsymbol{c}_t \sim \mathcal{D}(\cdot|x_t)$ *for all* t.

Next, we combine $\mathcal{B}_1$, $\mathcal{B}_3$ and $\mathcal{B}_4$. Although seemingly $\mathcal{B}_3$ is dominated by $\mathcal{B}_1$ since it has a worst regret bound under the same stochastic assumptions, in practice, $\mathcal{B}_3$ is computationally much faster than $\mathcal{B}_1$ (indeed, in total $\mathcal{B}_3$ only makes one call of the oracle while $\mathcal{B}_1$ makes $\widetilde{\mathcal{O}}(\sqrt{T})$ calls over T rounds), and therefore it can afford to use a more complicated policy class under the same time constraint. Similarly, although $\mathcal{B}_4$ dominates all other algorithms in terms of regret guarantee, its running time is linear in the number of policies and can only afford to use a very small policy class. For example, we can run $\mathcal{B}_4$ with a policy class of depth-5 decision trees, $\mathcal{B}_1$ with a larger policy class of depth-10 decision trees, and $\mathcal{B}_3$ with an even larger policy class of depth-20 decision trees. If the environment is easy in the sense that a depth-5 decision tree can predict well already, then $\mathcal{B}_4$ exploits this fact and achieves $\widetilde{\mathcal{O}}(\sqrt{T})$ regret without any stochastic assumption; otherwise, we still have $\mathcal{B}_1$ to provide $\widetilde{\mathcal{O}}(\sqrt{T})$ regret against a larger class, and $\mathcal{B}_3$ to provide $\widetilde{\mathcal{O}}(T^{3/4})$ regret against an even larger class, albeit under i.i.d. assumptions. Formally we have the following result:

Corollary 9 *Suppose we run* CORRAL *with three base algorithms:* ILOVETOCONBANDITS *with policy class* Θ_1, Epoch-Greedy *with policy class* Θ_3, *and* EXP4 *with policy class* Θ_4 *such that* $\Theta_4 \subset \Theta_1 \subset \Theta_3$. *If the learning rate* η *is set to* $1/\sqrt{KT \ln |\Theta_4|}$, *then we have:*

1. $\sup_{\theta \in \Theta_4} \mathbb{E}\left[\sum_{t=1}^T f_t(\theta_t, x_t) - f_t(\theta, x_t)\right] = \widetilde{\mathcal{O}}(\sqrt{KT \ln |\Theta_4|})$ *for adversarial* $x_t, \mathbf{c}_t$;
2. *the better of these two bounds if* $(x_t, \mathbf{c}_t)$ *are drawn i.i.d. from a fixed and unknown distribution:* $\sup_{\theta \in \Theta_1} \mathbb{E}\left[\sum_{t=1}^T f_t(\theta_t, x_t) - f_t(\theta, x_t)\right] = \widetilde{\mathcal{O}}(\sqrt{KT} \ln |\Theta_1| (\ln |\Theta_4|)^{-\frac{1}{2}})$, *and* $\sup_{\theta \in \Theta_3} \mathbb{E}\left[\sum_{t=1}^T f_t(\theta_t, x_t) - f_t(\theta, x_t)\right] = \widetilde{\mathcal{O}}\left(T^{\frac{3}{4}} K^{\frac{1}{2}} (\ln |\Theta_3|)^{\frac{3}{4}} (\ln |\Theta_4|)^{-\frac{1}{4}}\right)$.

5.1.2. CONVEX BANDITS

In convex bandit problems, $\Theta \subset \mathbb{R}^d$ is a compact convex set (assumed to have constant diameter after rescaling), side information x_t is usually empty, and the loss function f_t is assume to be convex

in θ. Without further assumptions, this is a rather difficult problem. Recent works (Bubeck et al., 2015; Bubeck and Eldan, 2016; Bubeck et al., 2016) make some important progress in this direction but unfortunately with very complicated and impractical algorithms. Here we consider two simpler and more practical algorithms. The first one is SCRiBLe (Abernethy et al., 2012) (denoted by $\mathcal{B}_1$) which was proposed under the assumption that the f_t's are linear functions. The second one is BGD from (Flaxman et al., 2005) (denoted by $\mathcal{B}_2$), which has a regret guarantee as long as the loss functions are Lipschitz. We show that both algorithms admit stability in Lemmas 20 and 21.

Again, direct application of Theorems 7 now leads to the following more adaptive algorithm:

Corollary 10 *Suppose we run* CORRAL *with two base algorithms:* SCRiBLe *and* BGD *with learning rate* $\eta = \frac{1}{d^{3/2}\sqrt{T}}$. *Assuming* $f_1, \ldots, f_T$ *are convex and L-Lipschitz, we have:*

1. $\sup_{\theta\in\Theta} \mathbb{E}\left[\sum_{t=1}^T f_t(\theta_t, x_t) - f_t(\theta, x_t)\right] = \widetilde{\mathcal{O}}\left(L^{\frac{2}{3}}(dT)^{\frac{5}{6}}\right)$;
2. *in addition, if the losses are linear, that is,* $f_t(\theta, x) = \langle \theta, \boldsymbol{c}_t\rangle$ *for some* $\boldsymbol{c}_t \in \mathbb{R}^d$, *then we have* $\sup_{\theta\in\Theta} \mathbb{E}\left[\sum_{t=1}^T f_t(\theta_t, x_t) - f_t(\theta, x_t)\right] = \widetilde{\mathcal{O}}(d^{\frac{3}{2}}\sqrt{T})$.

5.2. Robustness to Many Different Environments

Another application of CORRAL is to create an algorithm that works simultaneously under different environments and can select the correct model automatically. Although the dependence on the number of base algorithms is polynomial instead of logarithmic as is usually the case for model selection problems, there are still many scenarios where the number of models is relatively small and polynomial dependence is not a serious problem. We present several examples below.

5.2.1. MULTI-ARMED BANDITS

The classic K-armed bandit problem is simply the case where $\Theta = [K]$ and $f_t(\theta, x) = \langle \boldsymbol{c}_t, \boldsymbol{e}_\theta\rangle$. Although there exist algorithms (such as EXP3 (Auer et al., 2002b)) that guarantee the optimal $\widetilde{\mathcal{O}}(\sqrt{TK})$ regret even if the losses are generated adversarially, in practice, a Bayesian approach called Thompson Sampling (Thompson, 1933) is often used and known to perform well. However, like other Bayesian approaches, Thompson Sampling assumes a prior over the environments, and the regret guarantee is usually only meaningful when the prior is true.[5]

Nevertheless, with our ensemble approach, one can easily create an algorithm that works under different true priors. To present the results, we follow the analysis of (Russo and Van Roy, 2014). Suppose the loss vectors $\boldsymbol{c}_t$ are i.i.d samples of a distribution $\mathcal{D}$ which is itself drawn from a prior distribution $\mathcal{P}$ over a family of distributions. Let $\mu_{\mathcal{D}} = \mathbb{E}_{\boldsymbol{c}\sim\mathcal{D}}[\boldsymbol{c}]$ be the mean vector drawn from the prior and $q_i = \Pr_{\mathcal{D}\sim\mathcal{P}}\left(\mu_{\mathcal{D},i} \le \mu_{\mathcal{D},j},\ \ \forall\, j \in [K]\right)$ so that $\boldsymbol{q}$ is the distribution of the optimal arm. Let $H(\boldsymbol{q})$ be the entropy of $\boldsymbol{q}$. Then we have the following results (note that all expectations are taken with respect to the true prior in addition to all other randomness):

Corollary 11 *If we run* CORRAL *with* M *instances of Thompson Sampling, each of which uses a different prior* $\mathcal{P}_i$, *and the true prior* $\mathcal{P} = \mathcal{P}_{i^\star}$ *for some* $i^\star$, *then with* $\eta = \sqrt{\frac{M}{TK}}$ *we have* $\sup_{\theta\in\Theta} \mathbb{E}\left[\sum_{t=1}^T f_t(\theta_t, x_t) - f_t(\theta, x_t)\right] = \widetilde{\mathcal{O}}(\sqrt{MTK}H(\boldsymbol{q}^\star))$,*where* $\boldsymbol{q}^\star$ *is the distribution over the optimal arm induced by* $\mathcal{P}_{i^\star}$.

5. There is also worst-case analysis for Thompson Sampling; see for example (Agrawal and Goyal, 2012; Kaufmann et al., 2012).

5.2.2. OTHER EXAMPLES

We briefly mention some other examples without giving details. For contextual bandits, if we have different ways to represent the contexts, then each base algorithm can be any existing contextual bandit algorithm with a specific context representation and policy space. The master can then have good performance as long as one of these representations captures the problem well.

For stochastic linear bandits, $\Theta \subset \mathbb{R}^d$ is a compact convex set and $f_t(\theta, x) = \langle \theta, \boldsymbol{c}^* \rangle + \xi_t$ where $\boldsymbol{c}^* \in \mathbb{R}^d$ is fixed and unknown, and ξ_t is some zero-mean noise. Previous works have studied cases where $\boldsymbol{c}^*$ is assumed to admit some special structures, such as sparsity, group-sparsity and so on (see for example (Abbasi-Yadkori et al., 2012; Carpentier and Munos, 2012; Johnson et al., 2016)). One can then run CORRAL with different base algorithms assuming different structures of $\boldsymbol{c}^*$. Another related problem is generalized linear bandits, where $f_t(\theta, x) = \sigma(\langle \theta, \boldsymbol{c}^* \rangle) + \xi_t$ for some link function σ (such as the logistic function, exponential function and so on, see (Filippi et al., 2010)). It is clear that one can run CORRAL with different base algorithms using different link functions to capture more possibilities of the environments. In all these cases, the number of base algorithms is relatively small.

6. Conclusion and Open Problems

In this work, we presented a master algorithm which can combine a set of base algorithms and perform as well as the best of them in a very strong sense in the bandit setting. Two major applications of our approach were presented to illustrate how this master algorithm can be used to create more adaptive bandit algorithms in a black-box fashion.

There are two major open problems left in this direction. One is to improve the results of Theorem 7 so that the master can basically inherit the same regret bounds of all the base algorithms, i.e., Eq. (1) holds simultaneously for all base algorithms satisfying stability condition with $\alpha_i < 1$. Note that this is in general impossible (see (Lattimore and Szepesvári, 2016) for a lower bound in a special case), but it is not clear whether it is possible if we only care about the scaling with T while allowing worse dependence on other parameters. The current approach fails to achieve this mainly because each of these bounds requires a different tuning of the same learning rate η.

Another open problem is to improve the dependence on M, the number of base algorithms, from polynomial to logarithmic while keeping the same dependence on other parameters (or prove its impossibility). Logarithmic dependence on M can be achieved by using EXP4 as the master, but as was earlier discussed, this leads to poor dependence on other parameters.

Acknowledgments

The authors would like to thank John Langford for posing the question initially that stimulated this research. Most of the work was completed when Behnam Neyshabur was an intern at Microsoft Research.

References

Yasin Abbasi-Yadkori, David Pal, and Csaba Szepesvari. Online-to-confidence-set conversions and application to sparse stochastic bandits. In *AISTATS*, 2012.

Jacob D Abernethy, Elad Hazan, and Alexander Rakhlin. Interior-point methods for full-information and bandit online learning. *IEEE Transactions on Information Theory*, 58(7):4164–4175, 2012.

Jacob D Abernethy, Chansoo Lee, and Ambuj Tewari. Fighting bandits with a new kind of smoothness. In *Advances in Neural Information Processing Systems*, 2015.

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*, 2012.

Jean-Yves Audibert and Sébastien Bubeck. Regret bounds and minimax policies under partial monitoring. *Journal of Machine Learning Research*, 11(Oct):2785–2836, 2010.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Peter Auer and Chao-Kai Chiang. An algorithm with nearly optimal pseudo-regret for both stochastic and adversarial bandits. In *Proceedings of the 29th Annual Conference on Learning Theory (COLT)*, 2016.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002a.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002b.

Sbastien Bubeck and Nicol Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012. ISSN 1935-8237. doi: 10.1561/2200000024.

Sébastien Bubeck and Ronen Eldan. Multi-scale exploration of convex functions and bandit convex optimization. In *Proceedings of the 29th Annual Conference on Learning Theory (COLT)*, 2016.

Sébastien Bubeck and Aleksandrs Slivkins. The best of both worlds: Stochastic and adversarial bandits. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*, 2012.

Sébastien Bubeck, Nicolo Cesa-Bianchi, and Sham M. Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*, volume 23, 2012.

Sébastien Bubeck, Ofer Dekel, Tomer Koren, and Yuval Peres. Bandit convex optimization:$\sqrt{T}$ regret in one dimension. In *Proceedings of the 28th Annual Conference on Learning Theory (COLT)*, 2015.

Sébastien Bubeck, Ronen Eldan, and Yin Tat Lee. Kernel-based methods for bandit convex optimization. *arXiv preprint arXiv:1607.03084*, 2016.

Alexandra Carpentier and Rémi Munos. Bandit theory meets compressed sensing for high dimensional stochastic linear bandit. In *AISTATS*, 2012.

Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, May 1997.

Wei Chu, Lihong Li, Lev Reyzin, and Robert E Schapire. Contextual bandits with linear payoff functions. In *AISTATS*, volume 15, pages 208–214, 2011.

Uriel Feige, Tomer Koren, and Moshe Tennenholtz. Chasing ghosts: competing with stateful policies. In *Foundations of Computer Science (FOCS)*, pages 100–109. IEEE, 2014.

Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, pages 586–594, 2010.

Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394. Society for Industrial and Applied Mathematics, 2005.

Dylan Foster, Zhiyuan Li, Thodoris Lykouris, Karthik Sridharan, and Eva Tardos. Learning in games: Robustness of fast convergence. In *Advances in Neural Information Processing Systems*, 2016.

Nicholas Johnson, Vidyashankar Sivakumar, and Arindam Banerjee. Structured stochastic linear bandits. *arXiv preprint arXiv:1606.05693*, 2016.

Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *International Conference on Algorithmic Learning Theory*, pages 199–213. Springer, 2012.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*, pages 817–824, 2008.

Tor Lattimore and Csaba Szepesvári. Lower bounds for stochastic linear bandits. Blog posts at http://banditalgs.com/2016/10/20/lower-bounds-for-stochastic-linear-bandits, 2016.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. In *Foundations of Computer Science, 1989., 30th Annual Symposium on*, pages 256–261. IEEE, 1989.

Odalric-Ambrym Maillard and Rémi Munos. Adaptive bandits: Towards the best history-dependent strategy. In *AISTATS*, pages 570–578, 2011.

Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13. Siam, 1994.

Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. In *Proceedings of the 26th Annual Conference on Learning Theory (COLT)*, pages 993–1019, 2013.

Alexander Rakhlin and Karthik Sridharan. Bistro: An efficient relaxation-based method for contextual bandits. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of thompson sampling. *Journal of Machine Learning Research*, 2014.

Yevgeny Seldin and Aleksandrs Slivkins. One practical algorithm for both stochastic and adversarial bandits. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

Vasilis Syrgkanis, Haipeng Luo, Akshay Krishnamurthy, and Robert E Schapire. Improved regret bounds for oracle-based adversarial contextual bandits. In *Advances in Neural Information Processing Systems*, 2016.

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

Appendix A. More Examples of the Setting

In this section, we instantiate the notions of a decision, environment and bandit algorithm in our general setting for several concrete examples. These examples are meant to illustrate the generality intended by our setup, suggesting the potentially broad consequences of results obtained in it. We start with the most basic setting.

Example 2 (Multi-armed bandits) Multi-armed bandits (Lai and Robbins, 1985) is the simplest instantiation of our setup where there is no side-information x_t and the decision space is a set of K arms, that is, $\Theta = [K]$, and the loss function specifies the loss of pulling each arm so that $f_t(\theta, x) = \langle \boldsymbol{c}_t, \boldsymbol{e}_\theta \rangle$ for some $\boldsymbol{c}_t \in [0,1]^K$. There are two main types of environments for which algorithms have been developed for this problem:

Stochastic environment: In this case, the loss vectors $\boldsymbol{c}_t$ are independent random draws from some fixed distribution at each round, and the environment is fully characterized by this fixed distribution. Perhaps the most well-known algorithm in this setting is the UCB strategy (Auer et al., 2002a) which obtains an expected regret of at most $\widetilde{\mathcal{O}}(\sqrt{KT})$.[6]

Adversarial environment: A significantly harder setting is one where the loss vectors $\boldsymbol{c}_t$ are chosen arbitrarily by an adaptive adversary. That is, the environment is an arbitrary mapping from the history $(\theta_s, \boldsymbol{c}_s)_{s=1,\ldots,t-1}$ to the next loss vector $\boldsymbol{c}_t$. The EXP3 algorithm of Auer et al. (2002b) is an approach which gets an expected regret of $\widetilde{\mathcal{O}}(\sqrt{KT})$ in this harder setting.

6. We skip the discussion of more detailed gap-dependent bounds here.

There are several modifications and refinements of this basic setting which we skip over here. For instance, the stochastic environments have been further refined to when the expected losses of the best arm are substantially lower than the rest. In the adversarial setting, there are results that take advantage of the loss functions changing slowly, or having a budget on the total amount of change an adversary can induce, in order to get better results. Our subsequent results will potentially allow us to enjoy better guarantees in some of these special cases, while remaining robust in the worst case.

Example 3 (Contextual bandits, expanded version of Example 1) Contextual bandits (Langford and Zhang, 2008) is a generalization of the multi-armed bandit problem where the side information x_t (called a context) is non-empty. The learner's decision space Θ consists of a set of policies, where a policy maps contexts to a discrete set of actions, i.e. $\theta : X \mapsto [K]$. For instance, if the contexts are points in $\mathbb{R}^d$, a policy might be parametrized by a weight matrix $\boldsymbol{W} \in \mathbb{R}^{K \times d}$ so that $\theta(x) = \arg\max_{i \in [K]} \boldsymbol{W}_i x$ where $\boldsymbol{W}_i$ is the i-th row of $\boldsymbol{W}$. Different base algorithms can in general work with very different policy classes, which can be captured by different Θ_i in our setting. The loss function is again in the form $f_t(\theta, x) = \langle \boldsymbol{c}_t, \boldsymbol{e}_{\theta(x)} \rangle$ for some $\boldsymbol{c}_t \in [0,1]^K$. This problem has been studied under three main environments:

Stochastic contexts and losses: In the simplest instance, both the contexts x_t and losses $\boldsymbol{c}_t$ are drawn i.i.d. according to a fixed distribution, and the environment is characterized by this distribution. The Epoch-Greedy algorithm of Langford and Zhang (2008) suffers an expected regret of $\widetilde{\mathcal{O}}(T^{2/3})$ in this setting. The more recent work of Agarwal et al. (2014) has a better regret bound of $\widetilde{\mathcal{O}}(\sqrt{T})$, though at a significantly higher computational cost.

Adversarial contexts or losses: Several authors (Auer, 2002; Chu et al., 2011; Filippi et al., 2010) have studied environments where the contexts are chosen by an adversary, but the losses come from a fixed, parametric form such as $\mathbb{E}[\boldsymbol{c}_t] = \boldsymbol{W}^\star x_t$ where $\boldsymbol{W}^\star$ is some fixed, unknown weight matrix. Thus the environment is characterized by the adversarial strategy for picking the next context given the history, along with the conditional distribution of losses given the context. While this relaxes the i.i.d. assumption on the contexts in the first setting, it places a more restrictive model on the stochastic losses. Algorithms such as LinUCB and variants (Li et al., 2010; Chu et al., 2011) enjoy $\widetilde{\mathcal{O}}(\sqrt{T})$ regret in these settings. Other authors (Syrgkanis et al., 2016; Rakhlin and Sridharan, 2016) have studied settings where the contexts are i.i.d. from a fixed distribution, but the losses are picked in an adversarial manner, a strict generalization of the i.i.d. setting from above. Syrgkanis et al. (2016) have proposed an algorithm which suffers an expected regret of at most $\widetilde{\mathcal{O}}(T^{2/3})$ in this setting.

Adversarial contexts and losses: This is the hardest environment which was addressed in an early work of Auer et al. (2002b), who propose the EXP4 algorithm. This algorithm incurs an expected regret at most $\widetilde{\mathcal{O}}(\sqrt{T})$ in the most general setting, but is computationally inefficient.

Example 4 (Convex bandits) This setting is a different way of generalizing the multi-armed bandit problem, and was initiated by the work of Flaxman et al. (2005). In this setting, the side-information x_t is again empty. The decision space Θ is typically some convex, compact subset of $\mathbb{R}^d$ such as a ball in a chosen norm. The loss functions f_t are typically convex (in θ) functions with some added regularity conditions. The two most well-studied settings here are:

Adversarial linear functions: In this case, the loss functions are linear, that is $f_t(\theta, x) = \langle \boldsymbol{c}_t, \theta \rangle$ where each $\boldsymbol{c}_t \in \mathbb{R}^d$ is chosen by an adversary. Abernethy et al. (2012) present an algorithm for

this setting with an expected regret of $\widetilde{\mathcal{O}}(d^{3/2}\sqrt{T})$. Several authors have also improved the regret bound for specific sets Θ as well as when the loss vectors are i.i.d (e.g. (Bubeck et al., 2012)).

Adversarial convex functions: More generally, the loss functions can be general convex, Lipschitz-continuous functions of θ, with the environment described by the adversary's strategy for picking the next loss function given the history. Flaxman et al. (2005) develop an algorithm which incurs an expected regret at most $\mathcal{O}(dT^{3/4})$. These results have been refined in subsequent works making further smoothness and strong convexity assumptions on the loss functions, as well as to $\widetilde{\mathcal{O}}(d^{9.5}\sqrt{T})$ regret in the more general setting in a very recent work of Bubeck et al. (2016).

Thus we see that several prior works on learning with partial feedback are admissible under our model. We again highlight that this is only a very quick survey of a large body of literature, and we are omitting discussion of many other setups such as Lipschitz losses in a metric space, gap-dependent results in stochastic settings etc., all of which are also fully captured in our setting.

Appendix B. Proof of Theorem 2

Proof (*Sketch*) Consider a simple setting where there are 2 base algorithms $\mathcal{B}_1$ and $\mathcal{B}_2$, a total of 4 actions with $\Theta_1 = \Theta_2 = \{a_1, a_2, a_3, a_4\}$, and 2 possible environments $\mathcal{E}_1$ and $\mathcal{E}_2$, both of which assign (unknown) fixed losses to the actions so that each action deterministically yields its assigned loss every round. More specifically, in $\mathcal{E}_1$, the losses assigned to a_1 and a_2 are 0.1 and 0.2 or 0.2 and 0.1 with equal probability, and the losses assigned to a_3 and a_4 are 0.3 and 0.4 or 0.4 and 0.3 with equal probability. Similarly for $\mathcal{E}_2$, the situation are reversed so that a_1 and a_2 are always worse than a_3 and a_4.

Base algorithms $\mathcal{B}_1$ and $\mathcal{B}_2$ are two nearly-identical copies of the same simple algorithm designed specially for these two environments. Specifically, $\mathcal{B}_1$ pulls a_1 in the first round. If the observed loss is 0.1 or 0.3, it keeps playing a_1; if the observed loss is 0.2 or 0.4, it keeps playing a_2; any other observed loss will lead to uniformly random choices between a_1 and a_2 for the rest of the game. $\mathcal{B}_2$ is similar except it only plays a_3 and a_4 in the same fashion. Clearly, $\mathcal{B}_1$ has constant regret in $\mathcal{E}_1$ while $\mathcal{B}_2$ has constant regret in $\mathcal{E}_2$.

Now the claim is that for any master, its expected regret must be $\Omega(T)$ under either $\mathcal{E}_1$ or $\mathcal{E}_2$. This is because without the knowledge of which environment it is in, in the first round the master will inevitably follows the "wrong" base algorithm (that is, $\mathcal{B}_1$ in $\mathcal{E}_2$ or $\mathcal{B}_2$ in $\mathcal{E}_1$) with constant probability under either $\mathcal{E}_1$ or $\mathcal{E}_2$. Without loss of generality, assume $\mathcal{E}_2$ is the true environment and the master follows $\mathcal{B}_1$ and thus plays a_1 in the first round. It can then supply the right feedback to $\mathcal{B}_1$; however, it has no information about the loss of a_3, the action that $\mathcal{B}_2$ suggested, and therefore it fails to update $\mathcal{B}_2$ correctly and as a result $\mathcal{B}_2$ will choose the wrong action with constant probability for the rest of the rounds. This means that the master has no way of recovering from this error and picks up linear regret. ■

Appendix C. Proofs of Main Results

We start by stating a regret guarantee for LOG-BARRIER-OMD, whose proof mostly follows the standard analysis (see for example (Shalev-Shwartz, 2011)) except for the part involving the special

log barrier mirror map (which is also the part that is slightly different from (Foster et al., 2016)). We recall our earlier notation

$$h(y) = y - 1 - \ln(y), \quad \text{so that} \quad D_{\psi_t}(\boldsymbol{p}, \boldsymbol{q}) = \sum_{i=1}^{M} \frac{1}{\eta_{t,i}} h\left(\frac{p_i}{q_i}\right).$$

Lemma 12 LOG-BARRIER-OMD *ensures that for any $\boldsymbol{u} \in \Delta_M$, we have after T rounds*

$$\sum_{t=1}^{T} \langle \boldsymbol{p}_t - \boldsymbol{u}, \boldsymbol{\ell}_t \rangle \leq \sum_{t=1}^{T} \left(D_{\psi_t}(\boldsymbol{u}, \boldsymbol{p}_t) - D_{\psi_t}(\boldsymbol{u}, \boldsymbol{p}_{t+1}) \right) + \sum_{t=1}^{T} \sum_{i=1}^{M} \eta_{t,i} p_{t,i}^2 \ell_{t,i}^2 .$$

Proof For any $\boldsymbol{u} \in \Delta_M$, by the algorithm, direct calculations and the generalized Pythagorean theorem, we have for any t:

$$\begin{aligned}
\langle \boldsymbol{p}_t - \boldsymbol{u}, \boldsymbol{\ell}_t \rangle &= \langle \boldsymbol{p}_t - \boldsymbol{u}, \nabla \psi_t(\boldsymbol{p}_t) - \nabla \psi_t(\tilde{\boldsymbol{p}}_{t+1}) \rangle \\
&= D_{\psi_t}(\boldsymbol{u}, \boldsymbol{p}_t) - D_{\psi_t}(\boldsymbol{u}, \tilde{\boldsymbol{p}}_{t+1}) + D_{\psi_t}(\boldsymbol{p}_t, \tilde{\boldsymbol{p}}_{t+1}) \\
&\leq D_{\psi_t}(\boldsymbol{u}, \boldsymbol{p}_t) - D_{\psi_t}(\boldsymbol{u}, \boldsymbol{p}_{t+1}) + D_{\psi_t}(\boldsymbol{p}_t, \tilde{\boldsymbol{p}}_{t+1}).
\end{aligned}$$

It thus remains to prove $h\left(\frac{p_{t,i}}{\tilde{p}_{t+1,i}}\right) \leq \eta_{t,i}^2 p_{t,i}^2 \ell_{t,i}^2$. Notice that by the algorithm, we have $\frac{p_{t,i}}{\tilde{p}_{t+1,i}} = 1 + \eta_{t,i} p_{t,i} \ell_{t,i}$. Therefore by the definition of $h(y)$ and the fact $\ln(1+x) \geq x - x^2$ when $x \geq 0$ we arrive at

$$h\left(\frac{p_{t,i}}{\tilde{p}_{t+1,i}}\right) = \frac{p_{t,i}}{\tilde{p}_{t+1,i}} - 1 - \ln\left(\frac{p_{t,i}}{\tilde{p}_{t+1,i}}\right) = \eta_{t,i} p_{t,i} \ell_{t,i} - \ln\left(1 + \eta_{t,i} p_{t,i} \ell_{t,i}\right) \leq \eta_{t,i}^2 p_{t,i}^2 \ell_{t,i}^2,$$

which completes the proof. ■

Next, we use the above lemma along with the sophisticated learning rates schedule to give a bound on the master's regret to any base algorithm. Importantly, the bound includes a negative term that is in terms of $\rho_{T,i}$.

Lemma 13 CORRAL *ensures that for any $i \in [M]$, we have*

$$\mathbb{E}\left[\sum_{t=1}^{T} f_t(\theta_t, x_t) - f_t(\theta_t^i, x_t)\right] \leq \mathcal{O}\left(\frac{M \ln T}{\eta} + T\eta\right) - \mathbb{E}\left[\frac{\rho_{T,i}}{40\eta \ln T}\right]$$

Proof Fix all the randomness, let n_i be such that $\eta_{T,i} = \beta^{n_i}\eta$, where we assume $n_i \geq 1$ (the case $n_i = 0$ is trivial as one will see). Let $t_1, \ldots, t_{n_i}$ be the rounds where Line 10 is executed for base algorithm $\mathcal{B}_i$. Since $\frac{1}{\tilde{p}_{t_{n_i}+1,i}} > \rho_{t_{n_i},i} > 2\rho_{t_{n_i-1},i} > \ldots > 2^{n_i} M$ and $\frac{1}{\tilde{p}_{t,i}} \leq TM$ for any t by Line 8, we have $n_i \leq \log_2 T$.

It is clear that CORRAL is running LOG-BARRIER-OMD with $\boldsymbol{\ell}_t = \frac{f_t(\theta_t, x_t)}{\tilde{p}_{t,i_t}} \boldsymbol{e}_{i_t}$. We can therefore apply Lemma 12, focusing on the term $\sum_{t=1}^{T} D_{\psi_t}(\boldsymbol{u}, \boldsymbol{p}_t) - D_{\psi_t}(\boldsymbol{u}, \boldsymbol{p}_{t+1})$. By the fact that

Bregman divergence is non-negative and the learning rate $\eta_{t,j}$ for each $j \in [M]$ is non-decreasing in t, we have

$$\begin{aligned}
\sum_{t=1}^{T} D_{\psi_t}(\boldsymbol{u}, \boldsymbol{p}_t) - D_{\psi_t}(\boldsymbol{u}, \boldsymbol{p}_{t+1}) &\le D_{\psi_1}(\boldsymbol{u}, \boldsymbol{p}_1) + \sum_{t=1}^{T-1} \left(D_{\psi_{t+1}}(\boldsymbol{u}, \boldsymbol{p}_{t+1}) - D_{\psi_t}(\boldsymbol{u}, \boldsymbol{p}_{t+1})\right) \\
&= D_{\psi_1}(\boldsymbol{u}, \boldsymbol{p}_1) + \sum_{t=1}^{T-1} \sum_{j=1}^{M} \left(\frac{1}{\eta_{t+1,j}} - \frac{1}{\eta_{t,j}}\right) h\left(\frac{u_j}{p_{t+1,j}}\right) \\
&\qquad\qquad \text{(recall } h(y) \ge 0\text{)} \\
&\le D_{\psi_1}(\boldsymbol{u}, \boldsymbol{p}_1) + \left(\frac{1}{\eta_{t_{n_i}+1,i}} - \frac{1}{\eta_{t_{n_i},i}}\right) h\left(\frac{u_i}{p_{t_{n_i}+1,i}}\right) \\
&= D_{\psi_1}(\boldsymbol{u}, \boldsymbol{p}_1) + \frac{1-\beta}{\beta^{n_i}\eta} h\left(\frac{u_i}{p_{t_{n_i}+1,i}}\right) \\
&\le D_{\psi_1}(\boldsymbol{u}, \boldsymbol{p}_1) - \frac{1}{5\eta \ln T} h\left(\frac{u_i}{p_{t_{n_i}+1,i}}\right)
\end{aligned}$$

where the last step is by the fact $1 - \beta \le -\frac{1}{\ln T}$ and $\beta^{n_i} \le e^{\frac{\log_2 T}{\ln T}} \le 5$.

We now set $\boldsymbol{u} = (1 - \frac{1}{T})\boldsymbol{e}_i + \frac{1}{TM}\mathbf{1} \in \Delta_M$. Assuming $T \ge 2$ we have $u_i \ge 1 - \frac{1}{T} \ge \frac{1}{2}$ and $u_j \ge \frac{1}{TM}$ for all j. We can thus bound the first term as

$$D_{\psi_1}(\boldsymbol{u}, \boldsymbol{p}_1) = \sum_{j=1}^{M} \frac{1}{\eta_{1,j}} h\left(\frac{u_j}{p_{1,j}}\right) = \frac{1}{\eta} \sum_{j=1}^{M} h(Mu_j) = \frac{1}{\eta} \sum_{j=1}^{M} \ln\left(\frac{1}{Mu_j}\right) \le \frac{M \ln T}{\eta}.$$

For the second term, note that we have $\frac{u_i}{p_{t_{n_i}+1,i}} \ge \frac{1}{4\bar{p}_{t_{n_i}+1,i}} \ge 2^{n_i - 2} M \ge 1$ as long as $M \ge 2$. So with the facts that $h(y)$ is increasing when $y \ge 1$ and $\rho_{T,i} = \frac{2}{\bar{p}_{t_{n_i}+1,i}}$, we have

$$h\left(\frac{u_i}{p_{t_{n_i}+1,i}}\right) \ge h\left(\frac{1}{4\bar{p}_{t_{n_i}+1,i}}\right) = \frac{\rho_{T,i}}{8} - 1 - \ln\left(\frac{1}{4\bar{p}_{t_{n_i}+1,i}}\right) \ge \frac{\rho_{T,i}}{8} - 1 - \ln\left(\frac{TM}{4}\right)$$

and therefore

$$\sum_{t=1}^{T} D_{\psi_t}(\boldsymbol{u}, \boldsymbol{p}_t) - D_{\psi_t}(\boldsymbol{u}, \boldsymbol{p}_{t+1}) \le O\left(\frac{M \ln T}{\eta}\right) - \frac{\rho_{T,i}}{40\eta \ln T}.$$

Finally, with the definition of $\boldsymbol{\ell}_t$ and the facts

$$\langle \boldsymbol{p}_t - \boldsymbol{u}, \boldsymbol{\ell}_t \rangle \ge \langle (1-\gamma)\boldsymbol{p}_t - \boldsymbol{u}, \boldsymbol{\ell}_t \rangle = \langle \bar{\boldsymbol{p}}_t - \boldsymbol{e}_i, \boldsymbol{\ell}_t \rangle + \langle \boldsymbol{e}_i - \boldsymbol{u} - \frac{\gamma}{M}\mathbf{1}, \boldsymbol{\ell}_t \rangle \ge \langle \bar{\boldsymbol{p}}_t - \boldsymbol{e}_i, \boldsymbol{\ell}_t \rangle - \frac{2\ell_{t,i_t}}{TM}$$

and $\sum_{j=1}^{M} \eta_{t,j} p_{t,j}^2 \ell_{t,j}^2 \le \eta_{t,i_t} \frac{p_{t,i_t}^2}{\bar{p}_{t,i_t}^2} \le \eta \frac{\beta^{\log_2 T}}{(1-\gamma)^2} \le 20\eta$, together with Lemma 12, we arrive at

$$\sum_{t=1}^{T} \langle \bar{\boldsymbol{p}}_t - \boldsymbol{e}_i, \boldsymbol{\ell}_t \rangle \le O\left(\frac{M \ln T}{\eta} + T\eta\right) + \left(\sum_{t=1}^{T} \frac{2\ell_{t,i_t}}{TM}\right) - \frac{\rho_{T,i}}{40\eta \ln T}.$$

Note that the conditional expectation of $\ell_{t,j}$ with respect to the random draw of i_t is $f_t(\theta_t^j, x_t)$ for all $j \in [M]$ and the conditional expectation of ℓ_{t,i_t} is $\sum_{j=1}^M f_t(\theta_t^j, x_t) \leq M$. Also

$$\mathbb{E}[\langle \bar{\boldsymbol{p}}_t, \boldsymbol{\ell}_t \rangle] = \sum_{i=1}^M \bar{p}_{t,i} f_t(\theta_t^i, x_t) = \mathbb{E}[f_t(\theta_t, x_t)].$$

Taking the expectations then finishes the proof. ■

With the tool of Lemma 13, the proofs of Theorem 4, 5 and 7 are simply to decompose the regret of the master and to make use of the negative term to cancel the large regret of the base algorithm in some sense.

Proof [of Theorem 4, 5 and 7] We begin by splitting the regret into two parts, namely the regret of the master to $\mathcal{B}_i$ and the regret of $\mathcal{B}_i$ to a fixed point in Θ_i:

$$\begin{aligned}
&\sup_{\theta \in \Theta_i} \mathbb{E}\left[\sum_{t=1}^T f_t(\theta_t, x_t) - f_t(\theta, x_t)\right] \\
&= \mathbb{E}\left[\sum_{t=1}^T f_t(\theta_t, x_t) - f_t(\theta_t^i, x_t)\right] + \sup_{\theta \in \Theta_i} \mathbb{E}\left[\sum_{t=1}^T f_t(\theta_t^i, x_t) - f_t(\theta, x_t)\right] \\
&= \mathbb{E}\left[\sum_{t=1}^T f_t(\theta_t, x_t) - f_t(\theta_t^i, x_t)\right] + \sup_{\theta \in \Theta_i} \mathbb{E}\left[\sum_{t=1}^T f_t^i(\theta_t^i, x_t) - f_t^i(\theta, x_t)\right],
\end{aligned}$$

where we use the fact that the fake loss function f_t^i is an unbiased estimator of the true loss function f_t by construction. Theorem 4 then follows directly by the stability condition of $\mathcal{B}_i$ and Lemma 13.

Next, setting $\alpha_i = 1$ and $\eta = \min\left\{\frac{1}{40\mathcal{R}_i(T)\ln T}, \sqrt{\frac{M}{T}}\right\}$ we have

$$\begin{aligned}
\sup_{\theta \in \Theta_i} \mathbb{E}\left[\sum_{t=1}^T f_t(\theta_t, x_t) - f_t(\theta, x_t)\right] &\leq \widetilde{\mathcal{O}}\left(\frac{M}{\eta} + T\eta\right) - \mathbb{E}\left[\frac{\rho_{T,i}}{40\eta \ln T}\right] + \mathbb{E}[\rho_{T,i}]\mathcal{R}_i(T) \\
&\leq \widetilde{\mathcal{O}}\left(\sqrt{MT} + M\mathcal{R}_i(T)\right) - \mathbb{E}[\rho_{T,i}]\mathcal{R}_i(T) + \mathbb{E}[\rho_{T,i}]\mathcal{R}_i(T) \\
&= \widetilde{\mathcal{O}}\left(\sqrt{MT} + M\mathcal{R}_i(T)\right),
\end{aligned}$$

proving Theorem 5.

On the other hand, when $\alpha_i < 1$ we have

$$\begin{aligned}
\sup_{\theta \in \Theta_i} \mathbb{E}\left[\sum_{t=1}^T f_t(\theta_t, x_t) - f_t(\theta, x_t)\right] &\leq \widetilde{\mathcal{O}}\left(\frac{M}{\eta} + T\eta\right) + \mathbb{E}\left[\rho_{T,i}^{\alpha_i}\mathcal{R}_i(T) - \frac{\rho_{T,i}}{40\eta \ln T}\right] \\
&\leq \widetilde{\mathcal{O}}\left(\frac{M}{\eta} + T\eta + \mathcal{R}_i(T)^{\frac{1}{1-\alpha_i}}\eta^{\frac{\alpha_i}{1-\alpha_i}}\right)
\end{aligned}$$

where the last step is by maximizing the function $\rho^{\alpha_i}\mathcal{R}_i(T) - \frac{\rho}{40\eta \ln T}$ over $\rho > 0$. This proves Theorem 7. ■

Appendix D. Omitted Details for Section 5

To verify the stability condition for the base algorithms, by a standard doubling trick argument it suffices to verify the following weaker version of the condition where the algorithm knows a bound on the loss range ρ ahead of time:

Definition 14 *For some constant $\alpha \in (0,1]$ and non-decreasing function $\mathcal{R} : \mathbb{N}_+ \to \mathbb{R}_+$, an algorithm with decision space $\Theta_0 \subset \Theta$ is called $(\alpha, \mathcal{R})$-weakly-stable with respect to an environment $\mathcal{E}$ if its regret under $\mathcal{E}$ is $\mathcal{R}(T)$, and its regret under any induced environment $\mathcal{E}'$ is*

$$\sup_{\theta \in \Theta_0} \mathbb{E}\left[\sum_{t=1}^{T} f_t'(\theta_t, x_t) - f_t'(\theta, x_t)\right] \le \rho_T^{\alpha} \mathcal{R}(T) \tag{4}$$

where $\rho_T \ge \rho = \max_{t \in [T]} 1/p_t$ is given to the algorithm ahead of time, and all expectations are taken over the randomness of both $\mathcal{E}'$ and the algorithm.

In fact, here we can even incorporate the doubling trick nicely into CORRAL as described below. Suppose that the base algorithms take a loss-range parameter as input upon initialization. At the beginning, we initialize these algorithms with range parameter $\rho_{1,i} = 2M$. Moreover, we do an extra step in Line 10 of CORRAL: restart base algorithm $\mathcal{B}_i$ with range parameter $\rho_{t+1,i}$. It is clear that the losses that any instances of the base algorithms receive will not exceed their range parameter. We call each of these reruns of a base algorithm an instantiation of that algorithm.

Now the following theorem shows that the weak stability condition is enough to show all our results up to constants. (Note that instead of directly showing the stability defined in Definition 3, we prove Eq. (5) which is all we need from the stability condition).

Theorem 15 *If base algorithm $\mathcal{B}_i$ is $(\alpha_i, \mathcal{R}_i)$-weakly-stable with respect to an environment $\mathcal{E}$, then running* CORRAL *(with the above modification) under $\mathcal{E}$ ensures*

$$\sup_{\theta \in \Theta_i} \mathbb{E}\left[\sum_{t=1}^{T} f_t^i(\theta_t^i, x_t) - f_t^i(\theta, x_t)\right] \le \frac{2^{\alpha_i}}{2^{\alpha_i} - 1} \mathbb{E}\left[\rho_{T,i}^{\alpha_i}\right] \mathcal{R}_i(T) \tag{5}$$

Proof Reusing notation from Section C, let $t_1, \ldots, t_{n_i} < T$ be the rounds where Line 10 is executed. Also let $t_0 = 0$ and $t_{n_i+1} = T$ for notational convenience. Note that the entire game is divided into $n_i + 1 \le \lceil \log_2 T \rceil$ segments $[t_{k-1}+1, t_k]$ for $k = 1, \ldots, n_i + 1$ based on the restarting of $\mathcal{B}_i$. We then have by the weak stability condition and monotonicity of $\mathcal{R}_i$,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} f_t^i(\theta_t^i, x_t) - f_t^i(\theta, x_t)\right] \\
&= \sum_{k=1}^{\lceil \log_2 T \rceil} \Pr[n_i + 1 \ge k] \cdot \mathbb{E}\left[\sum_{t=t_{k-1}+1}^{t_k} f_t^i(\theta_t^i, x_t) - f_t^i(\theta, x_t) \,\middle|\, n_i + 1 \ge k\right] \\
&\le \mathcal{R}_i(T) \sum_{k=1}^{\lceil \log_2 T \rceil} \Pr[n_i + 1 \ge k] \cdot \mathbb{E}\left[\rho_{t_k,i}^{\alpha_i} \,\middle|\, n_i + 1 \ge k\right] \\
&= \mathcal{R}_i(T) \mathbb{E}\left[\sum_{k=1}^{n_i+1} \rho_{t_k,i}^{\alpha_i}\right].
\end{aligned}$$

Now let $S = \sum_{k=1}^{n_i+1} \rho_{t_k,i}^{\alpha_i}$ and note that

$$(2^{\alpha_i}-1)S = \sum_{k=1}^{n_i}\left((2\rho_{t_k,i})^{\alpha_i} - \rho_{t_{k+1},i}^{\alpha_i}\right) + (2\rho_{t_{n_i+1},i})^{\alpha_i} - \rho_{t_1,i}^{\alpha_i} \le (2\rho_{t_{n_i+1},i})^{\alpha_i} = 2^{\alpha_i}\rho_{T,i}^{\alpha_i}.$$

where we use the fact $2\rho_{t_k,i} \le \rho_{t_{k+1},i}$. This proves the theorem. ■

In the following subsections, we prove that weak stability holds for different algorithms discussed in Section 5 as listed in Table 1. Note that when we say that an algorithm satisfies the condition, we always mean that with appropriate parameters or even slight modifications it satisfies the condition. Moreover, for notation convenience we drop the subscript for ρ_T (which is overloading the notation ρ but they convey similar meanings anyway and there will not be confusion below), and define random variable s_t which is $1/p_t$ with probability p_t and 0 otherwise (p_t is defined in the induced environment $\mathcal{E}'$, not to be confused with $\boldsymbol{p}_t$ in CORRAL). Note that $\mathbb{E}[s_t] = 1$ and $\mathbb{E}[s_t^2] \le \rho$.

D.1. Contextual Bandits

Recall that we considered four algorithms: ILOVETOCONBANDITS, BISTRO+, Epoch-Greedy and EXP4, denoted by $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ and $\mathcal{B}_4$ respectively. Their stability parameters and the corresponding class of environments are listed in the following lemmas, followed by the proofs.

Lemma 16 *If $(x_1, \boldsymbol{c}_1), \ldots, (x_T, \boldsymbol{c}_T)$ are generated independently from a fixed and unknown distribution, then* ILOVETOCONBANDITS *is $(\frac{1}{2}, \widetilde{\mathcal{O}}(\sqrt{KT\ln|\Theta_1|}))$-weakly-stable.*

Lemma 17 *If $x_1, \ldots, x_T$ are generated independently from a fixed and known distribution, then* BISTRO+ *is $(\frac{1}{3}, \mathcal{O}((KT)^{\frac{2}{3}}(\ln|\Theta_2|)^{\frac{1}{3}}))$-weakly-stable.*

Lemma 18 *If $(x_1, \boldsymbol{c}_1), \ldots, (x_T, \boldsymbol{c}_T)$ are generated independently from a fixed and unknown distribution, then* Epoch-Greedy *is $(\frac{1}{3}, \widetilde{\mathcal{O}}(T^{\frac{2}{3}}\sqrt{K\ln|\Theta_3|}))$-weakly-stable.*

Lemma 19 *For any sequence $(x_1, \boldsymbol{c}_1), \ldots, (x_T, \boldsymbol{c}_T)$,* EXP4 *is $(\frac{1}{2}, \mathcal{O}(\sqrt{KT\ln|\Theta_4|}))$-weakly-stable.*

To ease notation we drop the subscript for the policy class Θ. We first point out that the context-loss sequence that the base algorithm faces in $\mathcal{E}'$ is $(x_1, \boldsymbol{c}_1'), \ldots, (x_T, \boldsymbol{c}_T')$ where $\boldsymbol{c}_t' = s_t\boldsymbol{c}_t$.

Proof [of Lemma 16] The only technicality here is that the original analysis of ILOVETOCONBANDITS (ILTCB for short) assumes an i.i.d. loss sequence, while the loss sequence $\boldsymbol{c}_t'$ has dependence since the sampling probability at a round t can depend on the entire history. This is, however, not a problem for the regret analysis as the martingale structure of the losses essential to their analysis is preserved.

Indeed, the only lemma involving concentration of the losses in their regret analysis is Lemma 11 in Agarwal et al. (2014). We reproduce the essential elements of that lemma here in order to establish the stability condition. Given the loss function $\boldsymbol{c}_t'$ by $\mathcal{E}'$, the ILTCB algorithm further creates a loss estimator $\boldsymbol{c}_t''$ so that

$$\boldsymbol{c}''_t(a) = c_t(a_t)s_t\frac{\mathbf{1}\{a = a_t\}}{Q_t(a)},$$

where a_t is the action picked by ILTCB and Q_t is the probability distribution over the actions induced at round t by ILTCB. Given a policy π, we now define the random variable $Z_t := \boldsymbol{c}''_t(\pi(x_t)) - \boldsymbol{c}_t(\pi(x_t))$. Letting H_{t-1} denote the history including everything prior to round t, it is easily seen that $\mathbb{E}[Z_t|H_{t-1}] = 0$. Furthermore, Z_t is measurable with respect to the filtration H_t so that it is a martingale difference sequence adapted to this filtration. It is clear that $|Z_t| \le \rho/\mu_t$ (where μ_t corresponds to $\min_a Q_t(a)$ as in their analysis). Furthermore, the conditional variance is bounded by

$$\mathbb{E}[Z_t^2 \mid H_{t-1}] \le \mathbb{E}\left[s_t^2|H_{t-1}\right]\mathbb{E}\left[\frac{\mathbf{1}\{\pi(x_t) = a_t\}}{Q_t(\pi(x_t))^2}\middle| H_{t-1}\right] \le \rho\mathcal{V}_t(\pi),$$

where the quantity $\mathcal{V}_t(\pi)$ is as defined in the analysis of Agarwal et al. (2014). Hence, we see that both the range and second moment of Z_t are scaled by ρ. Plugging this into the proof of their Lemma 11, we see that the RHS of their bound simply becomes

$$\rho\mathcal{V}_t(\pi)\lambda + \frac{\ln(4t^2|\Pi|/\delta)}{t\lambda} = \mathcal{V}_t(\pi)\lambda' + \frac{\rho\ln(4t^2|\Pi|/\delta)}{t\lambda'}$$

where $\lambda \in [0, \mu_t/\rho]$ and $\lambda' \in [0, \mu_t]$. For the rest of the proof, one can simply replace all $\ln(4t^2|\Pi|/\delta)$ with $\rho\ln(4t^2|\Pi|/\delta)$ and obtain the claimed bound. ■

Proof [of Lemma 17] BISTRO+ is a relaxation-based approach. For our setting, the relaxation REL remains similar: let $\epsilon_t \in \{-1, 1\}^K$ be a Rademacher random vector (i.e. each coordinate is an independent Rademacher random variable which is -1 or 1 with equal probability), $Z_t \in \{0, L\rho\}$ be a random variable which is $L\rho$ with probability $K/(L\rho)$ and 0 otherwise for some parameter L, and finally let $\xi_t = (x, \epsilon, Z)_{t+1:T}$. Then the new relaxation is defined as follows:

$$\text{REL}(I_{1:t}) = \mathbb{E}_{\xi_t}\left[R((x, \hat{c})_{1:t}, \xi_t)\right], \tag{6}$$

where

$$R((x, \hat{c})_{1:t}, \xi_t) = -\min_{\theta\in\Theta}\left(\sum_{\tau=1}^{t}\hat{c}_{\tau,\theta(x_\tau)} + \sum_{\tau=t+1}^{T}2\epsilon_{\tau,\theta(x_\tau)}Z_\tau\right) + (T-t)K/L,$$

$$\hat{c}_t = L\rho X_t\boldsymbol{e}_{\hat{y}_t}\mathbf{1}\{s_t \neq 0\},$$

$$X_t = \begin{cases} 1 & \text{with probability } \frac{c_{t,\hat{y}_t}}{L\rho p_t q_{t,\hat{y}_t}}, \\ 0 & \text{with the remaining probability,} \end{cases}$$

and I_t is the *information set* as in (Syrgkanis et al., 2016). Similarly, one can verify that this modified relaxation satisfies the following two admissible conditions:

$$\mathbb{E}_{x_t}\left[\min_{q_t\in\Delta_K}\max_{c_t\in[0,1]^K}\mathbb{E}_{\hat{y}_t\sim q_t, X_t, s_t}\left[s_t c_{t,\hat{y}_t} + \text{REL}(I_{1:t})\right]\right] \le \text{REL}(I_{1:t-1}),$$

$$\mathbb{E}_{\hat{y}_{1:T}\sim q_{1:T}, X_{1:T}, s_{1:T}}\left[\text{REL}(I_{1:T})\right] \ge -\min_{\theta\in\Theta}\sum_{t=1}^{T}c_{t,\theta(x_t)}$$

with the following admissible strategy:

$$q_t = \mathbb{E}_{\xi_t}\left[q_t(\xi_t)\right] \quad \text{where} \quad q_t(\xi_t) = \left(1 - \frac{K}{L}\right) q_t^*(\xi_t) + \frac{1}{L}\mathbf{1},$$

and

$$q_t^*(\xi_t) = \operatorname*{argmin}_{q \in \Delta_K} \max_{w_t \in D} \mathbb{E}_{\hat{c}_t \sim w_t}\left[\langle q, \hat{c}_t\rangle + R((x,\hat{c})_{1:t}, \xi_t)\right].$$

Here, D is a subset of all distributions over $\{\mathbf{0}, L\rho\boldsymbol{e}_1, \ldots, L\rho\boldsymbol{e}_K\}$ such that the mass for each non-zero vector is at most $1/(L\rho)$. Finally, the expected regret of this modified BISTRO+ is bounded by

$$\textsc{Rel}(\emptyset) \le \sqrt{TKL\rho \ln|\Theta|} + \frac{TK}{L},$$

which is $\mathcal{O}((TK)^{\frac{2}{3}}(\rho \ln|\Theta|)^{\frac{1}{3}})$ by choosing the optimal L. This proves the lemma. ■

Proof [of Lemma 18] Let $\bar{c}(\theta) = \mathbb{E}_{(x,\boldsymbol{c})}\left[c_{\theta(x)}\right]$ be the expected cost of a policy θ, $\theta^* = \arg\min_{\theta\in\Theta} \bar{c}(\theta)$ be the optimal policy, and $\theta_S^* = \arg\min_{\theta\in\Theta} \sum_{(x,\boldsymbol{c})\in S} c_{\theta(x)}$ be the empirically optimal policy with respect to a training set S.

For simplicity, consider the following simplest version of Epoch-Greedy. For the first T_0 rounds (for some parameter T_0 to be specified), actions are chosen uniformly at random. Then a training set $S = \{(x_t, \boldsymbol{c}_t'')\}_{t=1,\ldots,T_0}$ where $\boldsymbol{c}_t''$ is the usual importance weighted estimator of $\boldsymbol{c}_t'$ is constructed and fed to the ERM oracle to obtain θ_S^*. Finally θ_S^* is used for the rest of the game.

Now for a fixed policy θ, consider the random variable $Y_t = \bar{c}(\theta) - c''_{t,\theta(x_t)}$. It is clear that $|Y_t| \le K\rho$, $Y_1, \ldots, Y_{T_0}$ form a martingale difference sequence and

$$\mathbb{E}_t\left[Y_t^2\right] \le 2\left(\bar{c}^2(\theta) + \mathbb{E}_t\left[(c''_{t,\theta(x_t)})^2\right]\right) \le 2(1 + K\rho).$$

Therefore by Freedman's inequality for martingales (we use the version of (Agarwal et al., 2014, Lemma 9) and pick $\lambda = \min\{\frac{1}{K\rho}, \sqrt{\frac{\ln\frac{1}{\delta}}{T_0 K\rho}}\}$), we have with probability at least $1-\delta$,

$$\left|T_0\bar{c}(\theta) - \sum_{t=1}^{T_0} c''_{t,\theta(x_t)}\right| = \left|\sum_{t=1}^{T_0} Y_t\right| \le \mathcal{O}\left(K\rho\ln\tfrac{1}{\delta} + \sqrt{T_0 K\rho\ln\tfrac{1}{\delta}}\right)$$

A union bound then implies that with probability $1-\delta$ and notation $B = K\rho\ln\left(\frac{|\Theta|}{\delta}\right)/T_0$,

$$\left|\bar{c}(\theta_S^*) - \frac{1}{T_0}\sum_{t=1}^{T_0} c''_{t,\theta_S^*(x_t)}\right| \le \mathcal{O}\left(B + \sqrt{B}\right),$$

and thus with probability at least $1-2\delta$, we have by construction of θ_S^*

$$\bar{c}(\theta_S^*) - \bar{c}(\theta^*) \le \bar{c}(\theta_S^*) - \frac{1}{T_0}\sum_{t=1}^{T_0} c''_{t,\theta_S^*(x_t)} - \left(\bar{c}(\theta^*) - \frac{1}{T_0}\sum_{t=1}^{T_0} c''_{t,\theta^*(x_t)}\right) = \mathcal{O}\left(B + \sqrt{B}\right).$$

Therefore, setting $\delta = 1/T$ shows that the expected regret of Epoch-Greedy is at most

$$2 + T_0 + \mathcal{O}\left(\left(\frac{K\rho \ln(T|\Theta|)}{T_0} + \sqrt{\frac{K\rho \ln(T|\Theta|)}{T_0}}\right) T\right).$$

Further picking $T_0 = T^{\frac{2}{3}}\rho^{\frac{1}{3}}\sqrt{K\ln(T|\Theta|)}$ leads to

$$\mathcal{O}\left((T^{\frac{2}{3}}\rho^{\frac{1}{3}} + T^{\frac{1}{3}}\rho^{\frac{2}{3}})\sqrt{K\ln(T|\Theta|)}\right).$$

Finally note that we in fact only care about $\rho = \mathcal{O}(T)$ (see the proof of Lemma 13) and thus $\rho^{\frac{2}{3}} \le \mathcal{O}(\rho^{\frac{1}{3}}T^{\frac{1}{3}})$, which simplifies the above bound to $\mathcal{O}\left(T^{\frac{2}{3}}\rho^{\frac{1}{3}}\sqrt{K\ln(T|\Theta|)}\right)$. ■

Proof [of Lemma 19] By standard analysis (see Auer et al. (2002b)), the expected regret of EXP4 as a base algorithm is at most

$$\eta'\mathbb{E}\left[\sum_{t=1}^{T}\sum_{a=1}^{K}(c'_{t,a})^2\right] + \frac{\ln|\Theta|}{\eta'},$$

where η' is the internal learning rate parameter of EXP4. Noting that $\mathbb{E}[(c'_{t,a})^2] \le \mathbb{E}[s_t^2] \le \rho$ and picking the optimal η' complete the proof. ■

D.2. Convex Bandits

Lemma 20 *If $f_t(\theta, x) = \langle\theta, \boldsymbol{c}_t\rangle$ for some $\boldsymbol{c}_t \in \mathbb{R}^d$, then* SCRiBLe *is $(\frac{1}{2}, \widetilde{\mathcal{O}}(d^{\frac{3}{2}}\sqrt{T}))$-weakly-stable.*

Proof The linear loss that the base algorithm SCRiBLe faces is $\langle\theta, s_t\boldsymbol{c}_t\rangle$ for $t = 1, \ldots, T$. According to the proof of (Abernethy et al., 2012, Theorem 5.1), the expected regret of SCRiBLe as a base algorithm is at most

$$2\eta' d^2\mathbb{E}\left[\sum_{t=1}^{T}(\langle\theta_t^1, s_t\boldsymbol{c}_t\rangle)^2\right] + \frac{d\ln T}{\eta'}$$

where η' is the internal learning rate parameter of SCRiBLe and $\theta_1^1, \ldots, \theta_T^1$ are the decisions of SCRiBLe. Now noting that $\langle\theta_t^1, \boldsymbol{c}_t\rangle \le 1$ and $\mathbb{E}[s_t^2] \le \rho$ and picking the optimal η' give the final regret bound $\widetilde{\mathcal{O}}(d^{\frac{3}{2}}\sqrt{T\rho})$. ■

Lemma 21 *If $f_1, \ldots, f_T$ are L-Lipschitz, then* BGD *is $(\frac{1}{4}, \mathcal{O}(d\sqrt{L}T^{\frac{3}{4}}))$-weakly-stable.*

Proof BGD is essentially running a stochastic version of online gradient descent with gradient estimators $g_1, \ldots, g_T$. The key component in its analysis is Lemma 3.1 of (Flaxman et al., 2005), which gives a regret bound of order $\sqrt{\sum_{t=1}^{T}\mathbb{E}[\|g_t\|^2]} \le G\sqrt{T}$ for stochastic online gradient descent where $G > 0$ is a bound on $\|g_t\|$. When run with a master, BGD uses gradient estimators $s_1g_1, \ldots, s_Tg_T$. Since $\mathbb{E}[s_t] \le \rho$, it is clear that the regret bound for the corresponding stochastic

online gradient descent now becomes $\sqrt{\sum_{t=1}^{T} \mathbb{E}[\|s_t g_t\|^2]} \leq G\sqrt{T\rho}$. The rest of the analysis remains the same as (Flaxman et al., 2005). ■

D.3. Multi-Armed Bandits

Lemma 22 *If Thompson Sampling is run with the true prior $\mathcal{P}$, then it is $(\frac{1}{2}, \mathcal{O}(\sqrt{TKH(\boldsymbol{q})}))$-weakly-stable.*

Proof One slight modification here is that the master needs to pass s_t to the Thompson Sampling (TS) strategy in order to update the posterior distribution $\boldsymbol{u}_t$ of the loss $\boldsymbol{c}_t$ (and it is clear that when $s_t = 0$ no update happens). Let $\boldsymbol{c}'_t = s_t \boldsymbol{c}_t$, $\boldsymbol{q}_t$ be the posterior distribution of the optimal arm, and $v_{t,\theta} = \sum_{j=1}^{K} q_{t,j} \left(\mathbb{E}_t \left[c'_{t,\theta} | \theta^* = j \right] - \mathbb{E}_t \left[c'_{t,\theta} \right] \right)^2$ where $\mathbb{E}_t$ denotes the expectation conditional on everything up to time t and θ^* denotes the optimal arm. One key modification of the analysis of (Russo and Van Roy, 2014) is to realize that

$$v_{t,\theta} \leq \rho \sum_{j=1}^{K} q_{t,j} \left(\mathbb{E}_{c_t} \left[c_{t,\theta} | \theta^* = j \right] - \mathbb{E}_{c_t} \left[c_{t,\theta} \right] \right)^2 ,$$

which is then used to show that (with θ_t being the output of TS)

$$\sum_{t=1}^{T} \mathbb{E} \left[v_{t,\theta_t} \right] \leq \frac{\rho}{2} H(\boldsymbol{q})$$

by the exact same argument of the original analysis. The rest of the analysis remains the same. ■