On the Sample Complexity of Representation Learning in Multi-Task Bandits with Global and Local Structure

Alessio Russo*, **Alexandre Proutiere**

Division of Decision and Control Systems
KTH Royal Institute of Technology, Stockholm, SE
{alessior,alepro}@kth.se

Abstract

We investigate the sample complexity of learning the optimal arm for multi-task bandit problems. Arms consist of two components: one that is shared across tasks (that we call representation) and one that is task-specific (that we call predictor). The objective is to learn the optimal (representation, predictor)-pair for each task, under the assumption that the optimal representation is common to all tasks. Within this framework, efficient learning algorithms should transfer knowledge across tasks. We consider the best-arm identification problem with fixed confidence, where, in each round, the learner actively selects both a task, and an arm, and observes the corresponding reward. We derive instance-specific sample complexity lower bounds, which apply to any algorithm that identifies the best representation, and the best predictor for a task, with prescribed confidence levels. We devise an algorithm, OSRL-SC, that can learn the optimal representation, and the optimal predictors, separately, and whose sample complexity approaches the lower bound. Theoretical and numerical results demonstrate that OSRL-SC achieves a better scaling with respect to the number of tasks compared to the classical best-arm identification algorithm. The code can be found here https://github.com/rssalessio/OSRL-SC.

Introduction

Learning from previous tasks and transferring this knowledge may significantly improve the process of learning new tasks. This idea, at the core of transfer learning (Pan and Yang 2009; Skinner 1965; Woodworth and Thorndike 1901), lifelong learning (Thrun 1996) and multi-task learning (Baxter et al. 2000; Caruana 1995, 1997), has recently triggered considerable research efforts with applications in both supervised and reinforcement learning. Previous work on transfer and multi-task learning has mostly focused on batch learning problems (Lazaric 2012; Pan and Yang 2009), where when a task needs to be solved, a training dataset is directly provided. Online learning problems, where samples for a given task are presented to the learner sequentially, have been much less studied (Taylor and Stone 2009; Zhan and Taylor 2015).

In this paper, we consider a multi-task Multi-Armed Bandit (MAB) problem, where the objective is to find the optimal arm for each task using the fewest number of samples, while allowing to transfer knowledge across tasks. The problem is modelled as follows: in each round, the learner actively selects a task, and then selects an arm from a finite set of arms. Upon selecting an arm, the learner observes a random reward from an unknown distribution that represents the performance of her action in that particular task. To allow the transfer of knowledge across the various tasks, we study the problem for a simple, albeit useful model. We assume that the arms available to the learner consist of two components: one that is shared across tasks (that we call representation) and one that is task-specific (that we call predictor). Importantly, the optimal arms for the various tasks share the same representation. The benefit of using this model is that we can study the sample complexity of learning the best shared representation across tasks while learning the task-specific best action.

Contribution-wise[1], in this work we derive instance-specific sample complexity lower bounds satisfied by any (δ_G, δ_H)-PAC algorithm (such an algorithm identifies the best representation with probability at least $1-\delta_G$, and the best predictors with probability at least $1-\delta_H$). Again, our lower bounds can be decomposed into two components, one for learning the representation, and one for learning the predictors. We devise an algorithm, OSRL-SC, which can learn the optimal representation, and predictors, separately, and whose sample complexity approaches the lower bound, scaling at most as $H(G\log(1/\delta_G) + X\log(1/\delta_H))$. Technically, we also show a novel regularization technique to obtain unique allocations in best-arm identification problems with fixed confidence. Finally, we present numerical experiments to illustrate the gains in terms of sample complexity one may achieve by transferring knowledge across tasks. To the best of our knowledge, this paper is the first to study *how tasks should be scheduled* toward a sample-optimal instance-specific best-arm identification algorithm.

Related work. Multi-task learning has been investigated under different assumptions on the way the learner interacts with tasks. One setting concerns batch learning (often referred to as learning-to-learn), where the training datasets for all tasks are available at the beginning (Baxter et al. 2000;

*Corresponding author.

[1] All the proofs and numerical details can be found here https://arxiv.org/abs/2211.15129.

Maurer 2005, 2009; Maurer, Pontil, and Romera-Paredes 2013). The so-called batch-within-online setting considers that tasks arrive sequentially, but as soon as a task arrives, all its training samples are available (Balcan, Blum, and Vempala 2015; Pentina and Ben-David 2015; Pentina and Urner 2016; Alquier, Mai, and Pontil 2017).

Next, in the online multi-task learning (Agarwal, Rakhlin, and Bartlett 2008; Abernethy, Bartlett, and Rakhlin 2007; Dekel, Long, and Singer 2007; Cavallanti, Cesa-Bianchi, and Gentile 2010; Saha et al. 2011; Lugosi, Papaspiliopoulos, and Stoltz 2009; Murugesan et al. 2016; Yang et al. 2020), in each round, the learner observes a new sample for each task, which, in some cases, this may not be possible. Our framework is different as in each round the learner can only select a single task. This framework has also been considered in (Lazaric, Brunskill et al. 2013; Soare et al. 2014; Soare 2015; Alquier, Mai, and Pontil 2017; Wu, Wang, and Lu 2019), but typically there, the learner faces the same task for numerous consecutive rounds, and she cannot select the sequence of tasks. Also, note that the structure tying the reward functions of the various tasks together is different from ours. The structure tying the rewards of actions for various contexts is typically linear, and it is commonly assumed that there exists a common low-dimensional representation, or latent structure, to be exploited (Soare et al. 2014; Soare 2015; Deshmukh, Dogan, and Scott 2017; Kveton et al. 2017; Hao, Lattimore, and Szepesvari 2020; Lale et al. 2019; Yang et al. 2020; Lu, Meisami, and Tewari 2021), or that the reward is smooth across tasks and/or arms (Magureanu, Combes, and Proutiere 2014; Slivkins 2014). The aforementioned papers address scenarios where the context sequence is not controlled and investigate regret. Meta-learning is also closely connected to meta-learning (Cella, Lazaric, and Pontil 2020; Kveton et al. 2021; Azizi et al. 2022). In (Azizi et al. 2022) the authors investigate the problem of simple regret minimization in a fixed horizon setting when tasks are sampled i.i.d. from some unknown distribution.

Model and Assumptions

In this section, we describe the analytical model of the multi-task MAB problem considered, and describe the framework of best-arm identification for this class of multi-task MAB models.

Model. We consider multi-task MAB problems with a finite set $\mathcal{X}$ of X tasks. In each round $t \geq 1$, the learner chooses a task $x \in \mathcal{X}$ and an arm $(g, h) \in \mathcal{G} \times \mathcal{H}$. The components g and h are, respectively, referred to as the *representation* and the *predictor*. When in round t, $x(t) = x$ and the learner selects (g, h), she collects a binary reward $Z_t(x, g, h)$ of mean $\mu(x, g, h)$ (for the sake of the analysis we only analyze the binary case, although it can be extended to the Gaussian case as in (Garivier and Kaufmann 2016)). The rewards are $i.i.d.$ across rounds, tasks, and arms. Consequently, the system is characterized by $\mu = (\mu(x, g, h))_{x,g,h}$, which is unknown to the learner.

The main assumption made throughout the paper is that tasks share a common optimal representation $g^\star$: for any task $x \in \mathcal{X}$, there is a predictor $h_x^\star$ such that $(g^\star, h_x^\star)$ yields an optimal reward. Formally,

$$\forall x \in \mathcal{X}, \quad \mu(x, g^\star, h_x^\star) > \max_{(g,h) \neq (g^\star, h_x^\star)} \mu(x, g, h). \quad (1)$$

Moreover, note that *there is no assumption on the smoothness* of μ with respect to (x, g, h).

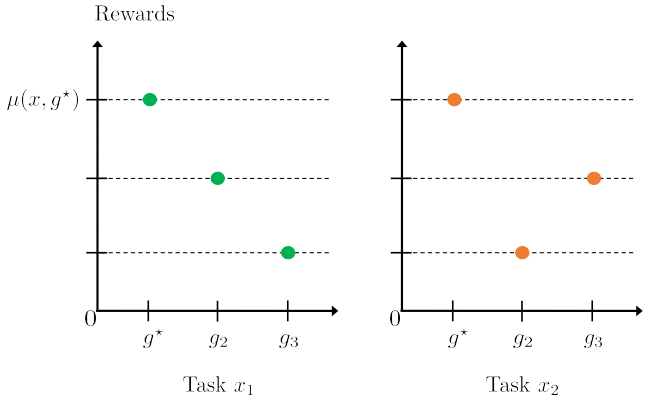

Figure 1: Example of two symmetric tasks x_1 and x_2, where learning the optimal representation $g^\star$ can be accelerated by considering both tasks, instead of focusing only on a single task. Task x_1 can be used to learn that g_3 is suboptimal, while task x_2 can be used to learn that g_2 is suboptimal.

This type of model represents the case where a learner can actively choose the task to execute (as if a generative model is available to the learner), and in this way maximize the learning efficiency by accurately picking tasks to reduce the sample complexity. Since the model is quite generic, it can be applied to a variety of problems where a collection of tasks have a local component, and a shared global component: (i) influence mechanisms with global/local groups; (ii) hyperparameters learning across multiple tasks; (iii) for advanced clinical trials, where, depending on the group of patients (tasks, that vary according to factors such as age, severity of the disease, etc.), different drugs and dosages can be used for inoculation (g and h).

Sample complexity. The objective of the learner is to devise a policy that learns the best arms $(g^\star, h_1^\star, \ldots, h_X^\star)$ with the least number of samples. Here, a policy π is defined as follows. Let $\mathcal{F}_t^\pi$ denote the σ-algebra generated by the observations made under π up to and including round t. Then π consists of (i) a sampling rule: in each round t, π selects a $\mathcal{F}_{t-1}$-measurable task $x^\pi(t)$ and an arm $(g^\pi(t), h^\pi(t))$; (ii) a stopping rule defined by τ, which is a stopping time w.r.t. the filtration $(\mathcal{F}_t)_{t \geq 1}$; (iii) a decision rule returning a $\mathcal{F}_\tau$-measurable estimated best arm for each task $(\hat{g}, \hat{h}_1, \ldots, \hat{h}_X)$. Then, the performance of a policy π is assessed through its PAC guarantees and its expected sample complexity $\mathbb{E}[\tau]$. PAC guarantees concern both learning $g^\star$ and $(h_1^\star, \ldots, h_X^\star)$. Denote by $\mathcal{M} = \{\mu : \exists (g^\star, h_1^\star, \ldots, h_X^\star) : \text{Eq. (1) holds}\}$ the set of systems where tasks share a common optimal representation. Then, we say

that π is (δ_G, δ_H)-PAC if for all $\mu \in \mathcal{M}$,

$$\mathbb{P}_\mu(\tau < \infty) = 1, \ \mathbb{P}_\mu\left(\hat{g} \neq g^\star\right) \leq \delta_G, \text{ and} \tag{2}$$

$$\mathbb{P}_\mu\left(\hat{h}_x \neq h_x^\star, \hat{g} = g^\star\right) \leq \delta_H, \quad \forall x \in \mathcal{X}. \tag{3}$$

Sample Complexity Lower Bound and the OSRL-SC Algorithm

This section is devoted to the best-arm identification problem for the model considered in this work. We first derive a lower bound for the expected sample complexity of any (δ_G, δ_H)-PAC algorithm, and then present an algorithm approaching this limit. In what follows, we let $\boldsymbol{\delta} = (\delta_G, \delta_H)$.

Sample Complexity Lower Bound

We begin by illustrating the intuition behind the sample complexity lower bound, and then state the lower bound theorem. To identify the optimal representation $g^\star$ in a task as quickly as possible, an algorithm should be able to perform some sort of *information refactoring*, *i.e.*, be able to use all the available information across tasks to estimate $g^\star$.

To illustrate this concept, we use the model illustrated in Fig. 1. In this model, there are only 2 tasks x_1, x_2, and 3 arms in $\mathcal{G}$ (and only 1 arm in $\mathcal{H}$, thus it can be ignored). For this model, to learn that g_3 is suboptimal, we should mainly sample task x_1, since the gap between the rewards of $g^\star$ and g_3 is the largest. Similarly, to learn that g_2 is suboptimal, we should mainly choose task x_2. Using the same task, to infer that g_2 and g_3 are suboptimal, would be less efficient. This observation also motivates why it is inefficient to consider tasks separately, even in the case where μ is highly non-smooth with respect to (x, g, h), and also motivates the expression of the sample complexity lower bound that we now present.

Sample complexity lower bound. Computing the sample complexity lower bound amounts to finding the lower bound of a statistical hypothesis testing problem, which is usually done by finding what is the most confusing model. In this case, the lower bound is given by the solution of the following optimization problem.

Theorem 1. The sample complexity $\tau_{\boldsymbol{\delta}}$ of any $\boldsymbol{\delta}$-PAC algorithm satisfies: $\mathbb{E}_\mu[\tau_{\boldsymbol{\delta}}] \geq K^\star(\mu, \boldsymbol{\delta})$ for any $\mu \in \mathcal{M}$, where $K^\star(\mu, \boldsymbol{\delta})$ is the value of the optimization problem[2]:

$$\min_{\eta} \quad \sum_{x,g,h} \eta(x, g, h) \tag{4}$$

$$\text{s.t.} \quad \min_{h \neq h_x^\star} f_\mu(\eta, x, h) \geq \mathrm{kl}(\delta_H, 1 - \delta_H) \quad \forall x, \tag{5}$$

$$\min_{\bar{g} \neq g^\star} \sum_x \min_{\bar{h}_x} \ell_\mu(\eta, \bar{g}, \bar{h}_x) \geq \mathrm{kl}(\delta_G, 1 - \delta_G), \tag{6}$$

where $\mathrm{kl}(a, b)$ is the KL divergence between two Bernoulli distributions of respective means a and b.

In the first constraint, $f_\mu(\eta, x, h) = (\eta(x, g^\star, h_x^\star) + \eta(x, g^\star, h)) I_{\alpha_{x,g^\star,h}}(\mu(x, g^\star, h_x^\star), \mu(x, g^\star, h))$ accounts for the difficulty of learning the best predictor $h_x^\star$ for each task x. The term $\alpha_{x,g,h} = \eta(x, g^\star, h_x^\star)/(\eta(x, g^\star, h_x^\star) + \eta(x, g, h))$ represents the proportion of time $(x, g^\star, h_x^\star)$ is picked over (x, g, h), while $I_\alpha(\mu_1, \mu_2), \alpha \in [0, 1]$, is a generalization of the Jensen-Shannon divergence, defined as

$$I_\alpha(\mu_1, \mu_2) \coloneqq \alpha \mathrm{kl}(\mu_1, d_\alpha(\mu_1, \mu_2)) + (1 - \alpha)\mathrm{kl}(\mu_2, d_\alpha(\mu_1, \mu_2)),$$

with $d_\alpha(\mu_1, \mu_2) \coloneqq \alpha\mu_1 + (1 - \alpha)\mu_2$.

In the second constraint, $\ell_\mu(\eta, \bar{g}, \bar{h}_x)$ accounts for the difficulty of learning the optimal $g^\star$. To define it, let the average reward over some subset of arms $\mathcal{C} \subseteq \mathcal{G} \times \mathcal{H}$ for a task x and allocation η to be defined as

$$m(x, \eta, \mathcal{C}) \coloneqq \frac{\sum_{(g,h) \in \mathcal{C}} \eta(x, g, h)\mu(x, g, h)}{\sum_{(g,h) \in \mathcal{C}} \eta(x, g, h)}. \tag{7}$$

Then, $\ell_\mu(\eta, \bar{g}, \bar{h}_x)$ is defined as:

$$\ell_\mu(\eta, \bar{g}, \bar{h}_x) \coloneqq \sum_{(g,h) \in \mathcal{U}^{\eta,\mu}_{x,\bar{g},\bar{h}_x}} \eta(x, g, h) \mathrm{kl}\left(\mu(x, g, h), m\left(x, \eta, \mathcal{U}^{\eta,\mu}_{x,\bar{g},\bar{h}_x}\right)\right), \tag{8}$$

where $\mathcal{U}^{\mu}_{x,\bar{g},\bar{h}_x}$ *is the set of confusing arms* for a task x and $m\left(x, \eta, \mathcal{U}^{\eta,\mu}_{x,\bar{g},\bar{h}_x}\right)$ represents the average reward of the confusing model (when $(\bar{g}, \bar{h}_x)$ is optimal for task x in the confusing model).

Confusing arms. The set $\mathcal{U}^{\eta,\mu}_{x,\bar{g},\bar{h}_x}$ is defined through $\mathcal{N}^{\mu}_{x,g,h;\bar{g},\bar{h}_x}$, which is the set of arms whose mean is bigger than $\mu(x, g, h)$ and that also include $(\bar{g}, \bar{h}_x)$, which is

$$\mathcal{N}^{\mu}_{x,g,h;\bar{g},\bar{h}_x} \coloneqq \{(g', h') : \mu(x, g', h') \geq \mu(x, g, h)\} \cup \{(\bar{g}, \bar{h}_x)\}.$$

Then, the set of confusing arms $\mathcal{U}^{\eta,\mu}_{x,\bar{g},\bar{h}_x}$ is defined as

$$\mathcal{U}^{\eta,\mu}_{x,\bar{g},\bar{h}_x} = \left\{(g, h) : \mu(x, g, h) \geq m\left(x, \eta, \mathcal{N}^{\mu}_{x,g,h;\bar{g},\bar{h}_x}\right)\right\} \cup \{(\bar{g}, \bar{h}_x)\}.$$

The set $\mathcal{U}^{\eta,\mu}_{x,\bar{g},\bar{h}_x}$ can be computed recursively. We start with a set that only contains $(g^\star, h_x^\star)$ and $(\bar{g}, \bar{h}_x)$. We compute the corresponding value of $m\left(x, \eta, \mathcal{U}^{\eta,\mu}_{x,\bar{g},\bar{h}_x}\right)$, and we add to the set $\mathcal{U}^{\eta,\mu}_{x,\bar{g},\bar{h}_x}$ the arm (g, h) with the highest mean not already in the set. We iterate until convergence. Fig. 2 provides an illustration of the set $\mathcal{U}^{\eta,\mu}_{x,\bar{g},\bar{h}_x}$.

We now provide the proof of Theorem 1.

Proof of Theorem 1. The proof relies on classical change-of-measure arguments, as those used in the classical MAB (Kaufmann, Cappé, and Garivier 2016). To simplify the notation, let $\tau = \tau_{\boldsymbol{\delta}}$, and further let $\eta(x, g, h) = \mathbb{E}_\mu[N_\tau(x, g, h)]$ at the stopping time τ, thus $\mathbb{E}_\mu[\tau] = \sum_{x,g,h} \eta(x, g, h)$. For any model $\mu \in \mathcal{M}$ we denote the optimal representation of μ by $g^\star(\mu)$ and the optimal set of predictors (associated to $g^\star(\mu)$) by $\boldsymbol{h}^\star(\mu) = (h_1^\star, \ldots, h_X^\star)(\mu)$. Whenever possible, we write $g^\star = g^\star(\mu)$ (similarly for $h_x^\star = h_x^\star(\mu), \forall x \in \mathcal{X}$).

We define the set of *confusing problems* as

$$\Lambda(\mu) \coloneqq \{\lambda \in \mathcal{M} : (g^\star, h_1^\star, \ldots, h_X^\star)(\lambda) \neq (g^\star, h_1^\star, \ldots, h_X^\star)(\mu)\}.$$

[2]Refer to the appendix for all the proofs.

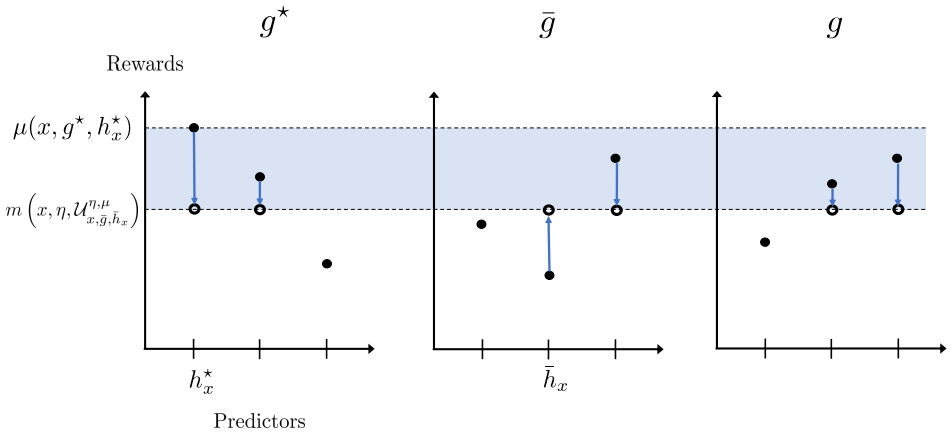

Figure 2: Example of the set $\mathcal{U}^{\eta,\mu}_{x,\bar{g},\bar{h}_x}$. All points in the shadowed area are in the set. All arms with average reward above $m\left(x, \eta, \mathcal{U}^{\eta,\mu}_{x,\bar{g},\bar{h}_x}\right)$ (including $(\bar{g}, \bar{h}_x)$) belong to $\mathcal{U}^{\eta,\mu}_{x,\bar{g},\bar{h}_x}$.

We split the analysis by considering two subsets of Λ, defined as follows:

$$\Lambda_1^\mu \coloneqq \{\lambda \in \Lambda(\mu) : g^\star(\lambda) = g^\star(\mu)\},$$
$$\Lambda_2^\mu \coloneqq \{\lambda \in \Lambda(\mu) : g^\star(\lambda) \neq g^\star(\mu)\}.$$

We now focus on the first subset Λ_1^μ, from which we derive the first constraint (5) of Theorem 1. Then, we focus on the second subset Λ_2^μ, from which follows the second constraint (6).

First constraint (5). We restrict our attention to Λ_1^μ. Define the set of confusing problems for task $x \in \mathcal{X}$ as

$$\Lambda_1^\mu(x) \coloneqq \{\lambda \in \Lambda_1^\mu : h_y^\star(\lambda) = h_y^\star(\mu), \forall y \in \mathcal{X} \setminus \{x\}\}.$$

Now, consider a (δ_G, δ_H)-PAC algorithm, and for a specific task $x \in \mathcal{X}$ define the event $\mathcal{E} = \{\hat{h}_x \neq h_x^\star, \hat{g} = g^\star\}$, where $\hat{h}_x$ and $\hat{g}$ denote respectively the estimated predictor for task x and the estimated optimal representation at the stopping time τ. Let then $\lambda \in \Lambda_1^\mu(x)$, be an alternative bandit model: the expected log-likelihood ratio L_τ at the stopping time τ of the observations under the two models μ and λ is given by

$$\mathbb{E}_\mu[L_\tau] = \sum_{(y,g,h) \in \mathcal{X}\times\mathcal{G}\times\mathcal{H}} \eta(y,g,h)\mathrm{kl}_{\mu|\lambda}(y,g,h),$$

and in view of the *transportation Lemma* 1 in (Kaufmann, Cappé, and Garivier 2016) and the definition of (δ_G, δ_H)-PAC algorithm, we can lower bound the previous quantity at the stopping time τ:

$$\begin{aligned}\mathbb{E}_\mu[L_\tau] &= \sum_{y,g,h} \eta(y,g,h)\mathrm{kl}_{\mu|\lambda}(y,g,h),\\ &\geq \mathrm{kl}(\mathbb{P}_\mu(\mathcal{E}), \mathbb{P}_\lambda(\mathcal{E})) = \mathrm{kl}(\delta_H, 1-\delta_H).\end{aligned}$$

We can get a tight lower bound by considering the worst possible model λ. To do so, first introduce the set $\Lambda_1^\mu(x,h) = \{\lambda \in \Lambda_1^\mu(x) : \lambda(x,g^\star,h) > \lambda(x,g^\star,h_x^\star)\}$, which is the set of confusing problems where the predictor $h_x^\star$ is not optimal in task x. Observe that one can write $\Lambda_1^\mu(x) = \cup_{h\neq h_x^\star}\Lambda_1^\mu(x,h)$. This rewriting allows us to derive the first constraint as follows:

$$\begin{aligned}\mathrm{kl}(\delta_H, 1-\delta_H) &\leq \inf_{\lambda\in\Lambda_1^\mu(x)} \sum_{y,g,h} \eta(y,g,h)\mathrm{kl}_{\mu|\lambda}(y,g,h),\\ &\overset{(a)}{=} \inf_{\lambda\in\Lambda_1^\mu(x)} \sum_{h} \eta(x,g^\star,h)\mathrm{kl}_{\mu|\lambda}(x,g^\star,h),\end{aligned}$$

where (a) follows from the fact that in λ we are changing the predictor of only one task x.

Then, the last term is equal to $\min_{h\neq h_x^\star} \inf_{\lambda\in\Lambda_1^\mu(x,h)} \Big[\eta(x,g^\star,h_x^\star)\mathrm{kl}_{\mu|\lambda}(x,g^\star,h_x^\star)$ $+\eta(x,g^\star,h)\mathrm{kl}_{\mu|\lambda}(x,g^\star,h)\Big] \overset{(b)}{=} \min_{h\neq h_x^\star}(\eta(x,g^\star,h_x^\star) + \eta(x,g^\star,h))I_{\alpha_{x,g^\star,h}}(\mu(x,g^\star,h_x^\star), \mu(x,g^\star,h))$ and (b) follows by solving the infimum problem as in Lemma 3 of (Garivier and Kaufmann 2016), and from the definition of generalized Shannon divergence.

Second constraint (6). Similarly to the previous case, consider a $(\delta_G, \delta_H)-$PAC algorithm and define the event $\mathcal{E} = \{\hat{g} \neq g^\star\}$: then, we can apply the transportation Lemma 1 in (Kaufmann, Cappé, and Garivier 2016) at the stopping time τ to obtain: for every $\lambda \in \Lambda_2^\mu$

$$\mathrm{kl}(\delta_G, 1-\delta_G) \leq \sum_{x,g,h} \eta(x,g,h)\mathrm{kl}_{\mu|\lambda}(x,g,h).$$

We consider subsets of Λ_2^μ defined as follows: for every $\bar{g} \in \mathcal{G}$ such that $\bar{g} \neq g^\star$ we define

$$\Lambda_2^\mu(\bar{g}) \coloneqq \{\lambda \in \Lambda_2^\mu : g^\star(\lambda) = \bar{g}\}.$$

Similarly, for all $(\bar{g}, \bar{\boldsymbol{h}}) \in (\mathcal{G} \setminus \{g^\star\}) \times \mathcal{H}^{\mathcal{X}}$, where $(\bar{g}, \bar{\boldsymbol{h}}) = (\bar{g}, \bar{h}_1, \ldots, \bar{h}_X)$, we also define

$$\Lambda_2^\mu(\bar{g}, \bar{\boldsymbol{h}}) \coloneqq \left\{\lambda \in \Lambda_2^\mu(\bar{g}) : (g^\star, h_1^\star, \ldots, h_X^\star)(\lambda) = (\bar{g}, \bar{\boldsymbol{h}})\right\}.$$

Thus, we observe

$$\Lambda_2^\mu = \cup_{\bar{g}\neq g^\star} \Lambda_2^\mu(g) = \cup_{\bar{g}\neq g^\star} \cup_{\bar{\boldsymbol{h}}\in\mathcal{H}^\mathcal{X}} \Lambda_2^\mu(\bar{g},\bar{\boldsymbol{h}}).$$

In conclusion, by minimizing the r.h.s. over the set of confusing models, *i.e.*, $\mathrm{kl}(\delta_G, 1-\delta_G) \leq \inf_{\lambda\in\Lambda_2^\mu} \sum_{x,g,h} \eta(x,g,h)\mathrm{kl}_{\mu|\lambda}(x,g,h)$, we obtain the following expression

$$\mathrm{kl}(\delta_G, 1-\delta_G) \leq \min_{\bar{g}\neq g^\star, \bar{\boldsymbol{h}}\in\mathcal{H}^\mathcal{X}} \sum_{x\in\mathcal{X}} \inf_{\lambda\in\Lambda_2^\mu(\bar{g},\bar{\boldsymbol{h}})} \sum_{g,h} \eta(x,g,h)\mathrm{kl}_{\mu|\lambda}(x,g,h),$$

which stems from the fact that $(\bar{g},\bar{\boldsymbol{h}})$ is fixed, for all the tasks. We conclude by observing that the right-hand side of the last inequality can be rewritten using Lemma (1) in the appendix. □

Discussion of Theorem 1. In Theorem 1 $\eta(x,g,h)$ can be interpreted as the minimal expected number of times any $\boldsymbol{\delta}$-PAC algorithm should select (x,g,h). Eq. (5) corresponds to the constraints on η one has to impose so that the algorithm learns the optimal predictor $h_x^\star$ for all x, while Eq. (6) is needed to ensure that the algorithm identifies the best representation $g^\star$ across all tasks. Both Eq. (5) and Eq. (6) define two convex sets in terms of η, and hence $K^\star(\mu,\boldsymbol{\delta})$ is the value of a convex program.

We wonder if it is possible to learn only the optimal representation $g^\star$, without learning the optimal predictors. In fact, observe that the constraints in Eq. (5) and Eq. (6) share the components of η that concern $g^\star$ only. We believe that actually separating the problem into two problems, one for each constraint, as formulated in the proposition below, yields a tight upper bound of $K^\star(\mu,\boldsymbol{\delta})$.

Proposition 1. We have $K^\star(\mu,\boldsymbol{\delta}) \leq K_H^\star(\mu,\delta_H) + K_G^\star(\mu,\delta_G)$, where $K_H^\star(\mu,\delta_H)$ (resp. $K_G^\star(\mu,\delta_G)$) is the value of the problem: $\min_{\eta\geq 0}\sum_{x,g,h}\eta(x,g,h)$ subject to (Eq. (5)) (resp. (Eq. (6))).

Note that $K_H^\star(\mu,\delta_H)$ scales as $HX\mathrm{kl}(\delta_H,1-\delta_H)$ (since the corresponding optimization problem is that obtained in a regular bandit problem for each task (Garivier and Kaufmann 2016), which scales as $H\mathrm{kl}(\delta_H,1-\delta_H)$ for each task). Now, to know how $K_G^\star(\mu,\delta_G)$ scales, we can further derive an upper bound of $K_G^\star(\mu,\delta_G)$.

Proposition 2. We have $K_G^\star(\mu,\delta_G) \leq L_G^\star(\mu,\delta_G)$ where $L_G^\star(\mu,\delta_G)$ is the value of the optimization problem: $\min\sum_{x,g,h}\eta(x,g,h)$ over $\eta \geq 0$ satisfying for all $\bar{g}\neq g^\star$, $\max_x \min_{\bar{h}_x}(\eta(x,g^\star,h_x^\star) + \eta(x,\bar{g},\bar{h}_x))I_{\alpha_{x,\bar{g},\bar{h}_x}}(\mu(x,g^\star,h_x^\star),\mu(x,\bar{g},\bar{h}_x)) \geq \mathrm{kl}(\delta_G,1-\delta_G)$.

One can show that $L_G^\star(\mu,\delta_G)$ scales as $GH\mathrm{kl}(\delta_G,1-\delta_G)$ (since, even with one task, we need to sample all GH arms to identify $g^\star$). To summarize, we have shown that the lower bound $K^\star(\mu,\boldsymbol{\delta})$ scales at most as $H(G\mathrm{kl}(\delta_G,1-\delta_G) + X\mathrm{kl}(\delta_H,1-\delta_H))$. The latter scaling indicates the gain in terms of sample complexity one can expect when exploiting the structure of $\mathcal{M}$, *i.e.*, a common optimal representation. Without exploiting this structure, identifying the best arm for each task would result in a scaling of $GHX\mathrm{kl}(\delta,1-\delta)$ for $\delta = \delta_G + \delta_H$.

Differences with classical best-arm identification. To better understand the lower bound in Theorem 1 it is instructive to compare it with a classical MAB problem.

Consider the best-arm identification problem in MAB with K arms. Then, the set of confusing problems is $\Lambda(\mu) = \{\lambda \in [0,1]^K : a^\star(\lambda) \neq a^\star(\mu)\}$, where $a^\star(\mu)$ denotes the optimal arm under μ, *i.e.*, $a^\star(\mu) = \arg\max_{a\in K}\mu(a)$. The sample complexity lower bound derived for such these models (Kaufmann, Cappé, and Garivier 2016; Garivier and Kaufmann 2016) exploits the fact that the set $\Lambda(\mu)$ can be written as $\Lambda(\mu) = \bigcup_{a\neq a^\star(\mu)}\Lambda_a(\mu)$, where

$$\Lambda_a(\mu) \coloneqq \left\{\lambda\in[0,1]^K : \lambda_a > \lambda_{a^\star(\mu)}\right\}.$$

Unfortunately, this way of rewriting the set of confusing problems cannot be used in our problem setting. The reason is that the constraint in $\Lambda_a(\mu)$ does not account for the model structure, *i.e.*, the optimal representation $g^\star$ needs to be the same across all the tasks (which is equivalent to imposing that a is optimal for all tasks). This fact also explains why the lower bound in Theorem 1 appears more complex than the one in (Garivier and Kaufmann 2016). In the appendix, we show how to account for this kind of structure.

Because of this difference, with the model specified in Eq. (1) the confusing parameter λ differs from μ for more than 2 arms (*i.e.*, we need to consider all arms in the set $\mathcal{U}^\eta_{x,\bar{g},\bar{h}_x}$, see also Lemma 1 in the appendix), whereas in classical MAB problems to learn that a is suboptimal, the confusing parameter $\lambda\in\Lambda_a(\mu)$ differs from μ only for arms a and $a^\star(\mu)$. In fact, in our model to identify that $(\bar{g},\bar{h}_x)$ is suboptimal, we need to consider an alternative model where only the average reward of the arms in the set $\mathcal{U}^{\eta,\mu}_{x,\bar{g},\bar{h}_x}$ changes.

Algorithm

We now present OSRL-SC (Algorithm 1), a $\boldsymbol{\delta}$-PAC algorithm whose expected sample complexity is asymptotically upper bounded by $K_G^\star(\mu,\delta_G) + K_H^\star(\mu,\delta_H)$. The algorithm proceeds in two phases: a first phase aimed at learning $g^\star$, and a second phase devoted to learning the optimal predictor for each task. At the end of the first phase, we have an estimate $\hat{g}$ of the best representation. In the second phase, for each task x, we use the optimal track-and-stop algorithm (Garivier and Kaufmann 2016) to identify $\hat{h}_x$, the best predictor associated to $\hat{g}$. In the remaining part of the section, we just describe the first phase.

A track-and-stop algorithm to learn $g^\star$. The lower bound describes the minimal expected numbers η of observations of the various tasks needed to learn $g^\star$. These numbers minimize $\sum_{x,g,h}\eta(x,g,h)$ over $\eta\geq 0$ satisfying (Eq. (6)). In other words, the sampling budget should be allocated according to the following distribution: $q^\star(\mu)\in\Sigma$, solving: $\max_{q\in\Sigma}\rho(q,\mu)$, where

$$\rho(q,\mu) = \min_{\bar{g}\neq g^\star}\sum_x \min_{\bar{h}_x}\ell_\mu(q,\bar{g},\bar{h}_x), \tag{9}$$

and Σ denotes the set of distributions over $\mathcal{X}\times\mathcal{G}\times\mathcal{H}$. We design an algorithm tracking this optimal allocation: it consists of (i) a sampling rule, (ii) a stopping rule, and a (iii)

Algorithm 1: OSRL-SC $(\delta_G, \delta_H, \sigma)$

Initialization.
$N_1(x,g,h), \hat{\mu}_1(x,g,h) \leftarrow 0, \forall (x,g,h) \in \mathcal{X} \times \mathcal{G} \times \mathcal{H}$
$t \leftarrow 1$
Phase 1. Learning $g^\star$
while $\max_{g \in G} \Psi_t(g) \leq \beta_t(\delta_G)$ **do**
  if $U_t = \emptyset$ and $\hat{\mu}_t \in \mathcal{M}$ **then**
    $(x(t), g(t), h(t)) \leftarrow \arg\max_{(x,g,h)} t q^\star_\sigma(x,g,h;\hat{\mu}_t) - N_t(x,g,h)$
  else
    $(x(t), g(t), h(t)) \leftarrow \arg\min_{(x,g,h)} N_t(x,g,h)$
  end if
  Select $(x(t), g(t), h(t))$, observe the corresponding reward and update statistics; $t \leftarrow t+1$
end while
return $\hat{g} = \arg\max_g \hat{\mu}_{\tau_G}(g)$
Phase 2. Learning $(\hat{h}^\star_1, \dots, \hat{h}^\star_X)$
For all task x, $\hat{h}_x \leftarrow$ [track-and-stop (Garivier and Kaufmann 2016) with arms $(\hat{g}, h)_{h \in \mathcal{H}}$ and confidence δ_H]
return $(\hat{g}, \hat{h}_1, \dots, \hat{h}_X)$

decision rule, described below.

(i) Sampling rule. We adapt the D-tracking rule introduced in (Garivier and Kaufmann 2016). The idea is to track $q^\star(\mu)$, the optimal proportion of times we should sample each (task, arm) pair. One important issue is that the solution to $\max_{q \in \Sigma} \rho(q,\mu)$ is not unique (this happens for example when two tasks are identical).

To solve this problem, we employ the following novel idea: we propose to regularize the optimization problem by tracking $q^\star_\sigma(\mu)$, the unique solution of

$$(P_\sigma) : \max_{q \in \Sigma} \rho(q,\mu) - \frac{1}{2\sigma}\|q\|_2^2, \quad \sigma > 0. \tag{10}$$

When σ is large, Berge's Maximum theorem (Berge 1963) implies that $q^\star_\sigma(\mu)$ approaches the set of solutions of $\max_{q \in \Sigma} \rho(q,\mu)$, and that the value $C_\sigma(\mu)$ of $\rho(q^\star_\sigma(\mu),\mu)$ converges to $K^\star_G(\mu,\delta_G)/\mathrm{kl}(\delta_G, 1-\delta_G)$. In what follows, we let $K^\star_{G,\sigma}(\mu,\delta_G) \coloneqq C_\sigma(\mu)\mathrm{kl}(\delta_G, 1-\delta_G)$.

Our D-tracking rule targets $q^\star_\sigma(\hat{\mu}_t)$, which is the unique maximizer of $\max_{q \in \Sigma} \rho(q,\hat{\mu}_t) - \frac{1}{2\sigma}\|q\|_2^2$, where $\rho(q,\hat{\mu}_t)$ for any $\hat{\mu}_t \in \mathcal{M}$ is defined as

$$\rho(q,\hat{\mu}_t) = \min_{\bar{g} \neq g^\star_t} \sum_x \min_{\bar{h}_x} \ell_{\hat{\mu}_t}(q, \bar{g}, \bar{h}_x). \tag{11}$$

More precisely, if the set of under-sampled tasks and arms $U_t = \{(x,g,h) \in \mathcal{X} \times \mathcal{G} \times \mathcal{H} : N_t(x,g,h) < \sqrt{t} - GHX/2\}$ is not empty, or when $\hat{\mu}_t \notin \mathcal{M}$, we select the least sampled (task, arm) pair. Otherwise, we track $q^\star_\sigma(\hat{\mu}_t)$, and select $\arg\max_{(x,g,h)} t q^\star_\sigma(x,g,h;\hat{\mu}_t) - N_t(x,g,h)$.

(ii) Stopping rule. We use Chernoff's stopping rule, which is formulated as a Generalized Likelihood Ratio Test (Chernoff 1959). The derivation of this stopping rule is detailed in appendix B. The stopping condition is $\max_{\tilde{g}} \Psi_t(\tilde{g}) > \beta_t(\delta_G)$, where the exploration threshold $\beta_t(\delta_G)$ needs to be appropriately chosen, and where

$$\Psi_t(\tilde{g}) = \min_{\bar{g} \neq \tilde{g}} \sum_x \min_{\bar{h}_x} \ell_{\hat{\mu}_t}(N_t, \bar{g}, \bar{h}_x).$$

(iii) Decision rule. The first phase ends at time τ_G, and $\hat{g}$ is chosen as the best empirical representation: $\hat{g} = \arg\max_g \hat{\mu}_t(g)$.

PAC and Sample Complexity Analysis

We now present the sample complexity upper bound for Algorithm 1. First, we outline the stopping rule used by the algorithm. Following (Kaufmann and Koolen 2018), we define $\phi : \mathbb{R}_+ \to \mathbb{R}_+$ as $\phi(x) = 2\tilde{p}_{3/2}\left(\frac{p^{-1}(1+x)+\ln(2\zeta(2))}{2}\right)$, where $\zeta(s) = \sum_{n \geq 1} n^{-s}$, $p(u) = u - \ln(u)$ for $u \geq 1$ and for any $z \in [1,e]$ and $x \geq 0$:

$$\tilde{p}_z(x) = \begin{cases} e^{1/p^{-1}(x)} p^{-1}(x) & \text{if } x \geq p^{-1}(1/\ln z), \\ z(x - \ln\ln z) & \text{otherwise.} \end{cases}$$

Then, the following theorem states that with a carefully chosen exploration threshold, the first phase of OSRL-SC returns the optimal representation w.p. greater than $1-\delta_G$.

Theorem 2. Let $\delta_G \in (0,1)$: for any sampling rule, Chernoff's stopping rule with threshold $\beta_t(\delta_G) = \beta_1(t) + \beta_2(\delta_G)$, where $\beta_1(t) = 3\sum_{x,g,h} \ln(1 + \ln(N_t(x,g,h)))$, and $\beta_2(\delta_G) = GHX\phi(\ln((G-1)/\delta_G)/XGH)$, ensures that for all $\mu \in \mathcal{M}$, $\mathbb{P}_\mu(\tau_G < \infty, \hat{g} \neq g^\star) \leq \delta_G$.

Proof. The proof can be seen as multi-task version of Proposition 12 in (Garivier and Kaufmann 2016) with an improved bound from (Kaufmann and Koolen 2018). Let $g^\star_t$ be the decision rule at time t. Since $\mathbb{P}_\mu(\tau_G < \infty, \hat{g} \neq g^\star) = \mathbb{P}_\mu(\exists t \in \mathbb{N} : g^\star_t \neq g^\star, \Psi_t(g^\star_t) > \beta_t(\delta_G))$, we can apply a union bound over the set $\mathcal{G} \setminus \{g^\star\}$ us to upper bound the probability of error as follows:

$$\mathbb{P}_\mu(\tau_G < \infty, \hat{g} \neq g^\star) \leq \sum_{\tilde{g} \neq g^\star} \mathbb{P}_\mu(\exists t : g^\star_t = \tilde{g}, \Psi_t(\tilde{g}) > \beta_t(\delta_G)).$$

Then, note that $\ell_{\hat{\mu}_t}(N_t, \bar{g}, \bar{h}_x)$ lower bounds $\sum_{(g,h) \in \mathcal{U}^{(t)}_{x,\bar{g},\bar{h}_x}} N_t(x,g,h)\mathrm{kl}(\hat{\mu}_t(x,g,h), \mu(x,g,h))$, with $\mathcal{U}^{(t)}_{x,\bar{g},\bar{h}_x} = \mathcal{U}^{N_t,\hat{\mu}_t}_{x,\bar{g},\bar{h}_x}$. Therefore, we derive that

$$\mathbb{P}_\mu(\tau_G < \infty, \hat{g} \neq g^\star) \leq \sum_{\tilde{g} \neq g^\star} \mathbb{P}_\mu\Big(\exists t \in \mathbb{N} : \sum_{x,g,h} N_t(x,g,h)\mathrm{kl}(\hat{\mu}_t(x,g,h), \mu(x,g,h)) > \beta_t(\delta_G)\Big).$$

We conclude by applying (Kaufmann and Koolen 2018, Thm. 14) with $x = \ln\left(\frac{G-1}{\delta_G}\right)$ and $\mathcal{S} = \mathcal{X} \times \mathcal{G} \times \mathcal{H}$. □

From (Garivier and Kaufmann 2016), the second phase of OSRL-SC also returns the optimal predictors for each task w.p. greater than $1-\delta_H$. Finally, in the next theorem, we show that OSRL-SC stops in finite time a.s., and that its expected sample complexity approaches $K^\star_G(\mu,\delta_G) + K^\star_H(\mu,\delta_H)$ for sufficiently small values of the risks δ_G, δ_H, and sufficiently large σ.

			Average	Confidence interval 97.5%	Min	Max	Std
$\delta = 0.1$	OSRL-SC	*Total*	21278.80	$\pm$430.37	5254.0	46876.0	6423.03
		Phase 1	**3578.38**	$\pm$**43.31**	**2163.0**	**7014.0**	**646.46**
		Phase 2	17700.42	$\pm$428.39	1554.0	43270.0	6393.44
	TAS		26456.83	$\pm$510.60	4544.0	54566.0	7620.35
$\delta = 0.05$	OSRL-SC	*Total*	22671.50	$\pm$420.40	6068.0	48184.0	6274.13
		Phase 1	**3651.99**	$\pm$**41.86**	**2358.0**	**6245.0**	**624.72**
		Phase 2	19019.51	$\pm$417.81	2207.0	45298.0	6235.60
	TAS		27735.38	$\pm$534.35	7675.0	58227.0	7974.87
$\delta = 0.01$	OSRL-SC	*Total*	25765.90	$\pm$436.13	8951.0	55809.0	6508.94
		Phase 1	**3829.56**	$\pm$**45.93**	**2358.0**	**7354.0**	**685.44**
		Phase 2	21936.34	$\pm$434.09	5398.0	52002.0	6478.57
	TAS		30970.94	$\pm$536.99	9538.0	70319.0	8014.26

Table 1: OSRL-SC vs TAS: Sample complexity results, over 1120 runs.

Theorem 3. If the exploration threshold of the first phase of OSRL-SC is chosen as in Theorem 2, then we have: $\mathbb{P}_\mu[\tau_G < \infty] = 1$ and $\mathbb{P}_\mu[\tau_H < \infty] = 1$ (where τ_H is the time at which the second phase ends). In addition, the sampling complexity of OSRL-SC satisfies: $\limsup_{\delta_G,\delta_H\to 0} \frac{\mathbb{E}_\mu[\tau]}{K^\star_{G,\sigma}(\mu,\delta_G)+K^\star_H(\mu,\delta_H)} \leq 1$, where $K^\star_{G,\sigma}(\mu,\delta_G) = C_\sigma(\mu)\mathrm{kl}(\delta_G, 1-\delta_G)$, with $C_\sigma(\mu) := \rho\left(q^\star_\sigma(\mu),\mu\right)^{-1}$.

Proof. The result follows from Theorem 7 (in the appendix) and Theorem 14 in (Garivier and Kaufmann 2016). The latter result upper bounds the expected duration of the second phase of OSRL-SC if we use, for this phase, a threshold rule $\beta_t(\delta_H)$ that is increasing in t and for which there exists constants $C, D > 0$ such that

$$\forall t \geq C, \forall \delta_H \in (0,1), \beta_t(\delta_H) \leq \ln\left(\frac{Dt}{\delta_H}\right).$$

For example, in the Bernoulli case, one can choose $\beta_t(\delta_H) = \log\left(\frac{2t(H-1)}{\delta_H}\right)$. Note that the sample complexity of the second phase can be rewritten as $\mathbb{E}_\mu[\tau_H] = \mathbb{E}_\mu[\tau_H|\hat{g}=g^\star]\mathbb{P}_\mu(\hat{g}=g^\star) + \mathbb{E}_\mu[\tau_H|\hat{g}\neq g^\star]\mathbb{P}_\mu(\hat{g}\neq g^\star)$, where $\mathbb{E}_\mu[\tau_H|\hat{g}=g]$ denotes the conditional expected sample complexity of the second phase, given that the first phase outputs $\hat{g}$. From the result of (Garivier and Kaufmann 2016), we know that

$$\limsup_{\delta_H\to 0} \frac{\mathbb{E}_\mu[\tau_H|\hat{g}=g^\star]}{K^\star_H(\mu,\delta_H;g^\star)} \leq 1,$$

where

$$K^\star_H(\mu,\delta_H;g^\star) := \sum_x T^\star(x,g^\star;\mu)\mathrm{kl}(\delta_H,1-\delta_H),$$

and $T^\star(x,g^\star;\mu)$ is defined as

$$T^\star(x,g^\star;\mu)^{-1} = \sup_{q\in\Sigma}\min_{h\neq\hat{h}^\star_x}\Big(\eta(x,g^\star,\hat{h}^\star_x) + \eta(x,g^\star,h)\Big) I_{\alpha_{x,g^\star,h}}(\mu(x,g^\star,\hat{h}^\star_x),\mu(x,g^\star,h)),$$

with $\hat{h}^\star_x = \arg\max_h \mu(x,g,h)$. For $g\neq g^\star$ we instead have

$$\limsup_{\delta_H\to 0} \frac{\mathbb{E}_\mu[\tau_H|\hat{g}=g]}{\mathrm{kl}(\delta_H,1-\delta_H)} \leq C,$$

for some positive constant C that depends on the threshold rule $\beta_t(\delta_H)$. Since the first phase of the algorithm is δ_G-PAC, we have:

$$\mathbb{E}_\mu[\tau_H] \leq \mathbb{E}_\mu[\tau_H|\hat{g}=g^\star]\mathbb{P}_\mu(\hat{g}=g^\star) + \mathbb{E}_\mu[\tau_H|\hat{g}\neq g^\star]\delta_G.$$

Therefore, we can conclude that for any positive δ_G we obtain

$$\limsup_{\delta_H\to 0} \frac{\mathbb{E}_\mu[\tau_H]}{K^\star_H(\mu,\delta_H;g^\star)\mathbb{P}_\mu(\hat{g}=g^\star)+\delta_G C(G-1)\mathrm{kl}(\delta_H,1-\delta_H)} \leq 1.$$

Hence, we obtain the result by letting $\delta_G\to 0$:

$$\limsup_{\delta_H,\delta_G\to 0} \frac{\mathbb{E}_\mu[\tau]}{K^\star_{G,\sigma}(\mu,\delta_G)+K^\star_H(\mu,\delta_H)} \leq 1.$$

□

Corollary 1. Additionally, due to Berge's theorem, since $\rho(q,\mu) - \frac{1}{2\sigma}\|q\|_2^2$ is continuous in q for each (σ,μ), we have: $\lim_{\sigma\to\infty} K^\star_{G,\sigma}(\mu,\delta_G) = K^\star_G(\mu,\delta_G)$.

Numerical Results

We analyze the performance of OSRL-SC, and compare it directly with TRACK AND STOP (TAS)(Garivier and Kaufmann 2016). We are interested in answering the following question: is it easier to learn the best representation by just focusing on one task, or should we consider multiple tasks at the same time?

Simulation setup. We consider 2 tasks (x_1, x_2), 3 representations (g_1, g_2, g_3) and 2 predictors (h_1, h_2). This setting is rather simple, although not trivial. Note that as the number of (task, arm) pairs decreases, we expect the gap between the two algorithms to decrease and thus favor TAS. Hence, considering examples with small numbers of tasks are informative about OSRL abilities to factor information across

Task x_1	(g_1, h_1)	(g_1, h_2)	(g_2, h_1)	(g_2, h_2)	(g_3, h_1)	(g_3, h_2)
$N_{\tau_G}(x_1)/\tau_G$	0.14	0.05	0.02	0.02	0.12	0.15
$q^\star_\sigma(x_1;\mu)$	0.18	$5 \cdot 10^{-3}$	$6 \cdot 10^{-4}$	$7 \cdot 10^{-4}$	0.13	0.18

Task x_2	(g_1, h_1)	(g_1, h_2)	(g_2, h_1)	(g_2, h_2)	(g_3, h_1)	(g_3, h_2)
$N_{\tau_G}(x_2)/\tau_G$	0.14	0.05	0.12	0.16	0.02	0.02
$q^\star_\sigma(x_2;\mu)$	0.18	$5 \cdot 10^{-3}$	0.13	0.18	$6 \cdot 10^{-4}$	$7 \cdot 10^{-4}$

Table 2: Analysis of OSRL-SC. Comparison of the optimal allocation vector $q^\star_\sigma(\mu)$ and the average proportion of arm pulls N_{τ_G}/τ_G at the stopping time.

tasks. The average rewards are

$$\underbrace{\begin{array}{c} \\ g_1 \\ g_2 \\ g_3 \end{array}\!\!\begin{array}{c} \begin{array}{cc} h_1 & h_2 \end{array} \\ \begin{pmatrix} 0.5 & 0.45 \\ 0.35 & 0.33 \\ 0.1 & 0.05 \end{pmatrix} \end{array}}_{\text{Average rewards for task } x_1}, \qquad \underbrace{\begin{array}{c} \\ g_1 \\ g_2 \\ g_3 \end{array}\!\!\begin{array}{c} \begin{array}{cc} h_1 & h_2 \end{array} \\ \begin{pmatrix} 0.5 & 0.45 \\ 0.1 & 0.05 \\ 0.35 & 0.33 \end{pmatrix} \end{array}}_{\text{Average rewards for task } x_2}$$

We set up tasks x_1 and x_2 so that they are very similar: actually, the only difference is that the 2nd and 3rd row of the above matrices are swapped. Therefore, it should not matter which task TAS picks, but, on the other hand, OSRL-SC should benefit from this small difference. Hence, for each simulation of TAS, we picked one task uniformly at random. Finally, we averaged results over 1120 runs.

Algorithms. We test TAS and OSRL-SC with various risks $\delta \in \{0.01, 0.05, 0.1\}$ (with $\delta = \delta_G = \delta_H$ for OSRL-SC). For TAS, we use the following threshold $\beta_t(\delta_G) = \log\left(\frac{2t(GH-1)}{\delta_g}\right)$. We tried the same threshold as in OSRL-SC, but it yielded worse results. For OSRL-SC, we set $\sigma = 10^5$. For the example considered, one can see that $\arg\max_{q \in \Sigma} \rho(q, \mu)$ has a unique maximizer. Therefore, σ will not influence the value of the lower bound if $\hat{\mu}_t$ is approximately close to μ, in norm. However, when $\hat{\mu}_t$ is visibly different from μ, then the value of $C_\sigma(\hat{\mu}_t)$ may be affected by the value of σ. We have not thoroughly explored different values of σ, but we may suggest that a value of $\sigma > 10^3$ is a safe choice.

We computed $q^\star_\sigma(\hat{\mu}_t)$ every 12 rounds (which is equal to GHX) to reduce the computational time (this is theoretically motivated in the appendix). Despite that, one needs to keep in mind that tracking a suboptimal, or wrong, reference vector $q^\star_\sigma$ may sensibly affect the sample complexity. We can also motivate this period update by the fact that $q^\star_\sigma(\hat{\mu}_t)$ in a small time interval does not change much, as shown numerically.

To compute $q^\star_\sigma(\hat{\mu}_t)$, in round $t+1$, we use as initial condition a convex combination of the previous solution and a uniform point in the probability simplex (with a factor 0.5). This is done to speed up the algorithm (for more details, refer to the appendix).

Comparison of OSRL-SC and TAS. In Table 1, we report the sample complexity of the two algorithms. In bold, we highlighted results for the first phase of OSRL-SC. Even if the number of representations is higher than the number of predictors, somewhat surprisingly, the first phase OSRL-SC seems, on average, very stable, with a small confidence interval (when compared to Phase 2 or TAS). It is worth observing that with the smallest number possible of tasks ($X = 2$), OSRL-SC manages to reduce the required number of rounds to identify the optimal representation, and the predictors, when compared to TAS. Furthermore, the first phase of OSRL-SC appears stable also when δ decreases. Between $\delta = 0.1$ and $\delta = 0.05$ there is a relative increase of average sample complexity of roughly 2% for OSRL-SC; between $\delta = 0.05$ and $\delta = 0.01$ we have that the average sample complexity for OSRL-SC has a relative increase of roughly 5%. Overall, these results indicate that OSRL-SC is able to re-factor information efficiently.

Analysis of OSRL-SC: First phase. To analyze the convergence of OSRL-SC, we focus on its first phase, specifically on the following quantities: $\hat{\mu}_t, q^\star_\sigma(\hat{\mu}_t)$ and $C^{-1}_\sigma(\hat{\mu}_t)$.

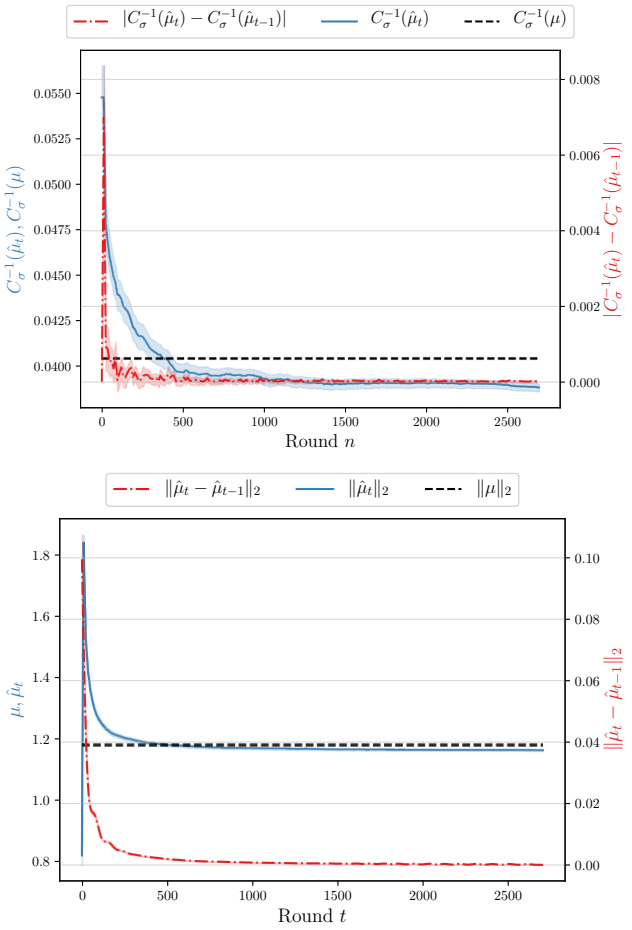

Figure 3: Analysis of $C^{-1}_\sigma(\hat{\mu}_t)$ and $\hat{\mu}_t$ under OSRL-SC. (a) Average dynamics of $C^{-1}_\sigma(\hat{\mu}_t)$, (b) Dynamics of $\hat{\mu}_t$. $\|\hat{\mu}_t - \hat{\mu}_{t-1}\|_2$ is normalized by $\sqrt{GHX}$ to show the average change of each component, and low-pass filtered using a 8-th order Butterworth filter with critical frequency $\omega_0 = 0.025$. The shadowed areas represent 97.5% confidence interval.

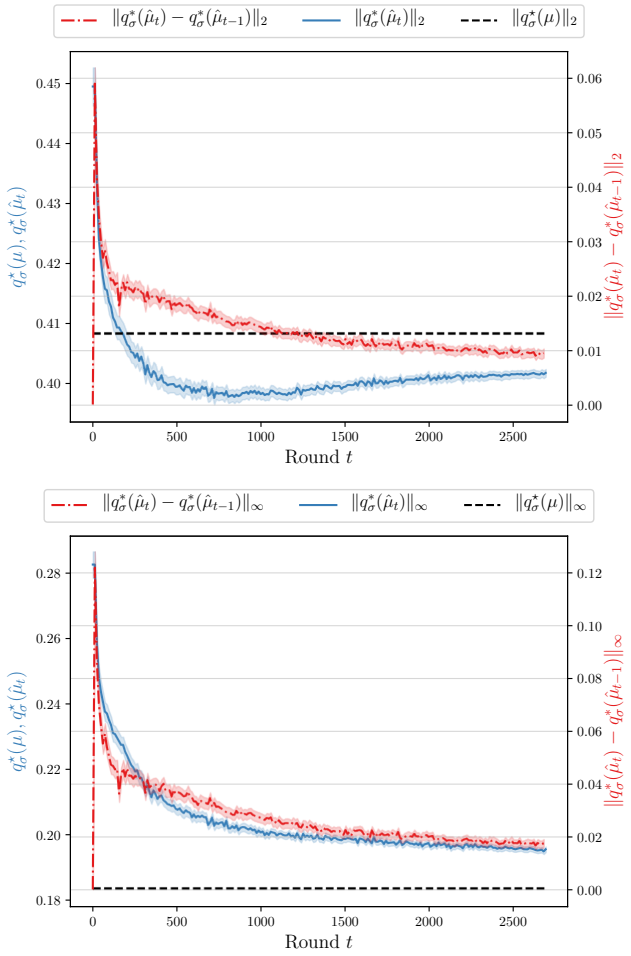

Figure 4: Analysis of $q^\star_\sigma(\hat{\mu}_t)$ under OSRL-SC. (a) Results using the 2-norm; (b) Results using the L^∞-norm. $\|q^\star_\sigma(\hat{\mu}_t) - q^\star_\sigma(\hat{\mu}_{t-1})\|_2$ is normalized by $\sqrt{GHX}$ to show the average change for each component. The shadowed areas represent 97.5% confidence interval.

We use the right y-axis of each plot to display the difference between the value in round t and round $t-1$ of the quantities considered.

Figure 3(b) shows how $\|\hat{\mu}_t - \hat{\mu}_{t-1}\|_2$ (normalized by $\sqrt{XGH}$) and $\|\hat{\mu}_t\|_2$ evolve over time. We clearly see that $\|\hat{\mu}_t\|_2$ quickly converges to some fixed value. This convergence appears in all the plots. Figure 3(a) shows the value of $C_\sigma^{-1}(\hat{\mu}_t)$, the true value $C_\sigma^{-1}(\mu)$, and the relative change of $C_\sigma^{-1}(\hat{\mu}_t)$ between two consecutive steps. We observe that the convergence rate of $\hat{\mu}_t$ dictates also the convergence of $C_\sigma^{-1}(\hat{\mu}_t)$. This suggests that we do not need to solve the lower bound optimization problem too often to update the target allocation, which helps reduce the computational complexity.

Figures 4(a) and (b) show 2 curves each: the left plot shows the 2-norm of $q^\star_\sigma(\hat{\mu}_t)$ and $q^\star_\sigma(\hat{\mu}_t) - q^\star_\sigma(\hat{\mu}_{t-1})$ (the latter normalized by $\sqrt{GHX}$), whilst the right plot shows the same signals computed using the L^∞-norm. In Figure 4(b), notice that the average absolute change in each component of the reference vector is very small, below 3% after few dozens of steps. Furthermore, we can see that this quantity has a convergence rate that is directly dictated by the convergence of $\hat{\mu}_t$ (even if its convergence rate is smaller). In Figure 4(a), observe that the relative difference between $q^\star_\sigma(\hat{\mu}_t)$ and $q^\star_\sigma(\mu)$ around $t = 2500$ is upper bounded by roughly $1/9$.

Finally, and importantly, in Table 2, we show the average proportion of arm pulls under OSRL-SC at the stopping time τ_G compared to the optimal allocation vector $q^\star_\sigma(\mu)$. It turns out that OSRL-SC follows accurately the optimal allocation. The algorithm picks the most informative arms in each task, *i.e.*, it adapts to the task. From this table, we can answer our initial question: to learn $g^\star$ as fast as possible, we need to use all tasks. Task 1 is used to learn that g_3 is suboptimal, and Task 2 is used to learn that g_2 is suboptimal. This is precisely what OSRL-SC is doing.

Conclusion

In this work, we analyzed knowledge transfer in stochastic multi-task bandit problems with finite arms, using the framework of best-arm identification with fixed confidence. We proposed OSRL-SC, an algorithm that transfers knowledge across tasks while approaching the sample complexity lower bound. We believe that this paper constitutes a sound starting point to study the transfer of knowledge in more general online multi-task learning problems. The limitation of this work is that we only consider models with a finite number of tasks and arms, which could limit their application in real life. Furthermore, our algorithm converges to an upper bound of the lower bound. Lastly, we think it would be interesting to study different structural assumptions (*e.g.* linearity) that tie reward functions across tasks together, or extend this work to multi-task reinforcement learning in MDPs.

Acknowledgements

This work was supported by the Swedish Foundation for Strategic Research through the CLAS project (grant RIT17-0046). In addition, the authors wish to thank Po-An Wang for all the extended feedback and insightful comments that he provided while the manuscript was being drafted.

References

Abernethy, J.; Bartlett, P.; and Rakhlin, A. 2007. Multitask learning with expert advice. In *International Conference on Computational Learning Theory*, 484–498. Springer.

Agarwal, A.; Rakhlin, A.; and Bartlett, P. 2008. Matrix regularization techniques for online multitask learning. *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2008-138*.

Alquier, P.; Mai, T. T.; and Pontil, M. 2017. Regret bounds for lifelong learning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 261–269.

Azizi, M.; Kveton, B.; Ghavamzadeh, M.; and Katariya, S. 2022. Meta-Learning for Simple Regret Minimization. *arXiv preprint arXiv:2202.12888*.

Balcan, M.-F.; Blum, A.; and Vempala, S. 2015. Efficient representations for lifelong learning and autoencoding. In *Conference on Learning Theory*, 191–210.

Baxter, J.; et al. 2000. A model of inductive bias learning. *J. Artif. Intell. Res.(JAIR)*, 12(149-198): 3.

Berge, C. 1963. Topological Spaces, Oliver and Boyed, Edinburg-London. *1st English edition*.

Caruana, R. 1995. Learning many related tasks at the same time with backpropagation. In *Advances in neural information processing systems*, 657–664.

Caruana, R. 1997. Multitask learning. *Machine learning*, 28(1): 41–75.

Cavallanti, G.; Cesa-Bianchi, N.; and Gentile, C. 2010. Linear algorithms for online multitask classification. *Journal of Machine Learning Research*, 11(Oct): 2901–2934.

Cella, L.; Lazaric, A.; and Pontil, M. 2020. Meta-learning with stochastic linear bandits. In *International Conference on Machine Learning*, 1360–1370. PMLR.

Chernoff, H. 1959. Sequential design of experiments. *The Annals of Mathematical Statistics*, 30(3): 755–770.

Dekel, O.; Long, P. M.; and Singer, Y. 2007. Online learning of multiple tasks with a shared loss. *Journal of Machine Learning Research*, 8(Oct): 2233–2264.

Deshmukh, A. A.; Dogan, U.; and Scott, C. 2017. Multi-task learning for contextual bandits. In *Advances in neural information processing systems*, 4848–4856.

Garivier, A.; and Kaufmann, E. 2016. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*, 998–1027.

Hao, B.; Lattimore, T.; and Szepesvari, C. 2020. Adaptive Exploration in Linear Contextual Bandit. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Kaufmann, E.; Cappé, O.; and Garivier, A. 2016. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1): 1–42.

Kaufmann, E.; and Koolen, W. 2018. Mixture martingales revisited with applications to sequential tests and confidence intervals. *arXiv preprint arXiv:1811.11419*.

Kveton, B.; Konobeev, M.; Zaheer, M.; Hsu, C.-w.; Mladenov, M.; Boutilier, C.; and Szepesvari, C. 2021. Meta-thompson sampling. In *International Conference on Machine Learning*, 5884–5893. PMLR.

Kveton, B.; Szepesvári, C.; Rao, A.; Wen, Z.; Abbasi-Yadkori, Y.; and Muthukrishnan, S. 2017. Stochastic low-rank bandits. *arXiv preprint arXiv:1712.04644*.

Lale, S.; Azizzadenesheli, K.; Anandkumar, A.; and Hassibi, B. 2019. Stochastic linear bandits with hidden low rank structure. *arXiv preprint arXiv:1901.09490*.

Lazaric, A. 2012. Transfer in reinforcement learning: a framework and a survey. In *Reinforcement Learning*, 143–173. Springer.

Lazaric, A.; Brunskill, E.; et al. 2013. Sequential transfer in multi-armed bandit with finite set of models. In *Advances in Neural Information Processing Systems*, 2220–2228.

Lu, Y.; Meisami, A.; and Tewari, A. 2021. Low-rank generalized linear bandit problems. In *International Conference on Artificial Intelligence and Statistics*, 460–468. PMLR.

Lugosi, G.; Papaspiliopoulos, O.; and Stoltz, G. 2009. Online multi-task learning with hard constraints. *arXiv preprint arXiv:0902.3526*.

Magureanu, S.; Combes, R.; and Proutiere, A. 2014. Lipschitz Bandits: Regret Lower Bounds and Optimal Algorithms. In *COLT, Barcelona, Spain, June 13-15, 2014*, volume 35, 975–999.

Maurer, A. 2005. Algorithmic stability and meta-learning. *Journal of Machine Learning Research*, 6(Jun): 967–994.

Maurer, A. 2009. Transfer bounds for linear feature learning. *Machine learning*, 75(3): 327–350.

Maurer, A.; Pontil, M.; and Romera-Paredes, B. 2013. Sparse coding for multitask and transfer learning. In *International Conference on Machine Learning*, 343–351.

Murugesan, K.; Liu, H.; Carbonell, J.; and Yang, Y. 2016. Adaptive smoothed online multi-task learning. In *Advances in Neural Information Processing Systems*, 4296–4304.

Pan, S. J.; and Yang, Q. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10): 1345–1359.

Pentina, A.; and Ben-David, S. 2015. Multi-task and lifelong learning of kernels. In *International Conference on Algorithmic Learning Theory*, 194–208. Springer.

Pentina, A.; and Urner, R. 2016. Lifelong learning with weighted majority votes. In *Advances in Neural Information Processing Systems*, 3612–3620.

Saha, A.; Rai, P.; DaumÃ, H.; Venkatasubramanian, S.; et al. 2011. Online learning of multiple tasks and their relationships. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 643–651.

Skinner, B. F. 1965. *Science and human behavior*. 92904. Simon and Schuster.

Slivkins, A. 2014. Contextual bandits with similarity information. *The Journal of Machine Learning Research*, 15(1): 2533–2568.

Soare, M. 2015. *Sequential resource allocation in linear stochastic bandits*. Ph.D. thesis, Université Lille 1-Sciences et Technologies.

Soare, M.; Alsharif, O.; Lazaric, A.; and Pineau, J. 2014. Multi-task linear bandits. In *NIPS2014 Workshop on Transfer and Multi-task Learning: Theory meets Practice*.

Taylor, M. E.; and Stone, P. 2009. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul): 1633–1685.

Thrun, S. 1996. Is learning the n-th thing any easier than learning the first? In *Advances in neural information processing systems*, 640–646.

Woodworth, R. S.; and Thorndike, E. 1901. The influence of improvement in one mental function upon the efficiency of other functions.(I). *Psychological review*, 8(3): 247.

Wu, Y.-S.; Wang, P.-A.; and Lu, C.-J. 2019. Lifelong Optimization with Low Regret. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 448–456.

Yang, J.; Hu, W.; Lee, J. D.; and Du, S. S. 2020. Impact of Representation Learning in Linear Bandits. In *International Conference on Learning Representations*.

Zhan, Y.; and Taylor, M. E. 2015. Online transfer learning in reinforcement learning domains. In *2015 AAAI Fall Symposium Series*.