# Model Selection in Reinforcement Learning with General Function Approximations

Avishek Ghosh*[1](✉) and Sayak Ray Chowdhury*[2]

[1] Halıcıoğlu Data Science Institute (HDSI), UC San Diego, USA
[2] Boston University, USA
a2ghosh@ucsd.edu, sayak@bu.edu

**Abstract.** We[3] consider model selection for classic Reinforcement Learning (RL) environments – Multi Armed Bandits (MABs) and Markov Decision Processes (MDPs) – under general function approximations. In the model selection framework, we do not know the function classes, denoted by $\mathcal{F}$ and $\mathcal{M}$, where the true models – reward generating function for MABs and and transition kernel for MDPs – lie, respectively. Instead, we are given $M$ nested function (hypothesis) classes such that true models are contained in at-least one such class. In this paper, we propose and analyze efficient model selection algorithms for MABs and MDPs, that *adapt* to the smallest function class (among the nested $M$ classes) containing the true underlying model. Under a separability assumption on the nested hypothesis classes, we show that the cumulative regret of our adaptive algorithms match to that of an oracle which knows the correct function classes (i.e., $\mathcal{F}$ and $\mathcal{M}$) a priori. Furthermore, for both the settings, we show that the cost of model selection is an additive term in the regret having weak (logarithmic) dependence on the learning horizon $T$.

**Keywords:** Model Selection · Bandits · Reinforcement Learning

## 1 Introduction

We study the problem of *model selection* for Reinforcement Learning problems, which refers to choosing the appropriate hypothesis class, to model the mapping from actions to expected rewards. We choose two particular frameworks—(a) Multi-Armaed Bandits (MAB) and (b) markov Decision Processes (MDP). Specifically, we are interested in studying the model selection problems for these frameworks without any function approximations (like linear, generalized linear etc.). Note that, the problem of model selection plays an important role in applications such as personalized recommendations, autonomous driving, robotics as we explain in the sequel. Formally, a family of nested hypothesis classes $\mathcal{H}_f$, $f \in \mathcal{F}$ is specified, where each class posits a plausible model for mapping actions to expected rewards. Furthermore, the family $\mathcal{F}$ is totally ordered, i.e., if $f_1 \leqslant f_2$, then $\mathcal{H}_{f_1} \subseteq \mathcal{H}_{f_2}$. It is assumed that the true model is contained in

[3] * Avishek Ghosh and Sayak Ray Chowdhury contributed equally.

at least one of these specified families. Model selection guarantees then refer to algorithms whose regret scales in the complexity of the *smallest hypothesis class containing the true model*, even though the algorithm was not aware apriori.

Multi-Armed Bandits (MAB) [7] and Markov decision processes (MDP) [25] are classical frameworks to model a reinforcement learning (RL) environment, where an agent interacts with the environment by taking successive decisions and observe rewards generated by those decisions. One of the objectives in RL is to maximize the total reward accumulated over multiple rounds, or equivalently minimize the *regret* in comparison with an optimal policy [7]. Regret minimization is useful in several sequential decision-making problems such as portfolio allocation and sequential investment, dynamic resource allocation in communication systems, recommendation systems, etc. In these settings, there is no separate budget to purely explore the unknown environment; rather, exploration and exploitation need to be carefully balanced.

Optimization over large domains under restricted feedback is an important problem and has found applications in dynamic pricing for economic markets [6], wireless communication [8] and recommendation platforms (such as Netflix, Amazon Prime). Furthermore, in many applications (e.g., robotics, autonomous driving), the number of actions and the observable number of states can be very large or even infinite, which makes RL challenging, particularly in generalizing learnt knowledge across unseen states and actions. For example, the game of Go has a state space with size $3^{361}$, and the state and action spaces of certain robotics applications can even be continuous. In recent years, we have witnessed an explosion in the RL literature to tackle this challenge, both in theory (see, e.g., [4, 10, 18, 22, 29]), and in practice (see, e.g., [21, 31]).

In the first part of the paper, we focus on learning an unknown function $f^* \in \mathcal{F}$, supported over a compact domain, via online noisy observations. If the function class $\mathcal{F}$ is known, the optimistic algorithm of [26] learns $f^*$, yielding a regret that depends on *eluder dimension* (a complexity measure of function classes) of $\mathcal{F}$. However, in the applications mentioned earlier, it is not immediately clear how one estimates $\mathcal{F}$. Naive estimation techniques may yield an unnecessarily big $\mathcal{F}$, and as a consequence, the regret may suffer. On the other hand, if the estimated class, $\widehat{\mathcal{F}}$ is such that $\widehat{\mathcal{F}} \subset \mathcal{F}$, then the learning algorithm might yield a linear regret because of this infeasibility. Hence, it is important to estimate the function class properly, and here is where the question of model selection appears. The problem of model selection is formally stated as follows—we are given a family of $M$ hypothesis classes $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \ldots \subset \mathcal{F}_M$, and the unknown function $f^*$ is assumed to be contained in the family of nested classes. In particular, we assume that $f^*$ lies in $\mathcal{F}_{m^*}$, where $m^*$ is unknown. Model selection guarantees refer to algorithms whose regret scales in the complexity of the *smallest model class containing the true function $f^*$, i.e., $\mathcal{F}_{m^*}$*, even though the algorithm is not aware of that a priori.

In the second part of the paper, we address the model selection problem for generic MDPs without funcction approximation. The most related work to ours is by [4], which proposes an algorithm, namely `UCRL-VTR`, for model-based RL

without any structural assumptions, and it is based on the upper confidence RL and value-targeted regression principles. The regret of `UCRL-VTR` depends on the *eluder dimension* [26] and the *metric entropy* of the corresponding family of distributions $\mathcal{P}$ in which the unknown transition model $P^*$ lies. In most practical cases, however, the class $\mathcal{P}$ given to (or estimated by) the RL agent is quite pessimistic; meaning that $P^*$ actually lies in a small subset of $\mathcal{P}$ (e.g., in the game of Go, the learning is possible without the need for visiting all the states [27]). We are given a family of $M$ nested hypothesis classes $\mathcal{P}_1 \subset \mathcal{P}_2 \subset \ldots \subset \mathcal{P}_M$, where each class posits a plausible model class for the underlying RL problem. The true model $P^*$ lies in a model class $\mathcal{P}_{m*}$, where $m^*$ is unknown apriori. Similar to the functional bandits framework, we propose learning algorithms whose regret depends on the *smallest model class containing the true model* $P^*$.

The problem of model selection have received considerable attention in the last few years. Model selection is well studied in the contextual bandit setting. In this setting, minimax optimal regret guarantees can be obtained by exploiting the structure of the problem along with an eigenvalue assumption [9, 12, 15] We provide a comprehensive list of recent works on bandit model selection in Section 1.2. However, to the best of our knowledge, this is the first work to address the model selection question for generic (functional) MAB without imposing any assumptions on the reward structure.

In the RL framework, the question of model selection has received little attention. In a series of works, [23, 24] consider the corralling framework of [2] for contextual bandits and reinforcement learning. While the corralling framework is versatile, the price for this is that the cost of model selection is multiplicative rather than additive. In particular, for the special case of linear bandits and linear reinforcement learning, the regret scales as $\sqrt{T}$ in time with an additional multiplicative factor of $\sqrt{M}$, while the regret scaling with time is strictly larger than $\sqrt{T}$ in the general contextual bandit. These papers treat all the hypothesis classes as bandit arms, and hence work in a (restricted) partial information setting, and as a consequence explore a lot, yielding worse regret. On the other hand, we consider all $M$ classes at once (full information setting) and do inference, and hence explore less and obtain lower regret.

Very recently, [20] study the problem of model selection in RL with function approximation. Similar to the *active-arm elimination* technique employed in standard multi-armed bandit (MAB) problems [11], the authors eliminate the model classes that are dubbed misspecified, and obtain a regret of $\mathcal{O}(T^{2/3})$. On the other hand, our framework is quite different in the sense that we consider model selection for RL with *general* transition structure. Moreover, our regret scales as $\mathcal{O}(\sqrt{T})$. Note that the model selection guarantees we obtain in the linear MDPs are partly influenced by [15], where model selection for linear contextual bandits are discussed. However, there are a couple of subtle differences: (a) for linear contextual framework, one can perform pure exploration, and [15] crucially leverages that and (b) the contexts in linear contextual framework is assumed to be i.i.d, whereas for linear MDPs, the contexts are implicit and depend on states, actions and transition probabilities.

### 1.1 Our Contributions

In this paper, our setup considers *any general* model class (for both MAB and MDP settings) that are totally bounded, i.e., for arbitrary precision, the metric entropy is bounded. Note that this encompasses a significantly larger class of environments compared to the problems with function approximation. Assuming nested families of reward function and transition kernels, respectively for MABs and MDPs, we propose adaptive algorithms, namely *Adaptive Bandit Learning* (`ABL`) and *Adaptive Reinforcement Learning* (`ARL`). Assuming the hypothesis classes are separated, both `ABL` and `ARL` construct a test statistic and thresholds it to identify the correct hypothesis class. We show that these *simple* schemes achieve the regret of $\tilde{\mathcal{O}}(d^* + \sqrt{d^*\mathbb{M}^*T})$ for MABs and $\tilde{\mathcal{O}}(d^*H^2 + \sqrt{d^*\mathbb{M}^*H^2T})$ for MDPs (with episode length $H$), where $d^*_{\mathcal{E}}$ is the *eluder dimension* and $\mathbb{M}^*$ is the *metric entropy* corresponding to the smallest model classes containing true models ($f^*$ for MAB and $P^*$ for MDP). The regret bounds show that both `ABL` and `ARL` adapts to the true problem complexity, and the cost of model section is only $\mathcal{O}(\log T)$, which is minimal compared to the total regret.

**Notation** For a positive integer $n$, we denote by $[n]$ the set of integers $\{1, 2, \ldots, n\}$. For a set $\mathcal{X}$ and functions $f, g : \mathcal{X} \to \mathbb{R}$, we denote $(f-g)(x) := f(x) - g(x)$ and $(f-g)^2(x) := (f(x)-g(x))^2$ for any $x \in \mathcal{X}$. For any $P : \mathcal{Z} \to \Delta(\mathcal{X})$, we denote $(Pf)(z) := \int_{\mathcal{X}} f(x)P(x|z)dx$ for any $z \in \mathcal{Z}$, where $\Delta(\mathcal{X})$ denotes the set of signed distributions over $\mathcal{X}$.

### 1.2 Related Work

**Model Selection in Online Learning:** Model selection for bandits are only recently being studied [9,13]. These works aim to identify whether a given problem instance comes from contextual or standard setting. For linear contextual bandits, with the dimension of the underlying parameter as a complexity measure, [12,15] propose efficient algorithms that adapts to the *true* dimension of the problem. While [12] obtains a regret of $\mathcal{O}(T^{2/3})$, [15] obtains a $\mathcal{O}(\sqrt{T})$ regret (however, the regret of [15] depends on several problem dependent quantities and hence not instance uniform). Later on, these guarantees are extended to the generic contextual bandit problems without linear structure [16, 19], where $\mathcal{O}(\sqrt{T})$ regret guarantees are obtained. The algorithm `Corral` was proposed in [2], where the optimal algorithm for each model class is casted as an expert, and the forecaster obtains low regret with respect to the best expert (best model class). The generality of this framework has rendered it fruitful in a variety of different settings; see, for example [2, 3].

**RL with Function Approximation:** Regret minimization in RL under function approximation is first considered in [22]. It makes explicit model-based assumptions and the regret bound depends on the eluder dimensions of the models. In contrast, [32] considers a low-rank linear transition model and propose a model-based algorithm with regret $\mathcal{O}(\sqrt{d^3H^3T})$. Another line of work parameterizes the *Q-functions* directly, using state-action feature maps, and develop model-free algorithms with regret $\mathcal{O}(\texttt{poly}(dH)\sqrt{T})$ bypassing the need for fully

learning the transition model [17, 30, 35]. A recent line of work [29, 33] generalize these approaches by designing algorithms that work with general and neural function approximations, respectively.

## 2 Model Selection in Functional Multi-armed Bandits

Consider the problem of sequentially maximizing an unknown function $f^* : \mathcal{X} \to \mathbb{R}$ over a compact domain $\mathcal{X} \subset \mathbb{R}^d$. For example, in a machine learning application, $f^*(x)$ can be the validation accuracy of a learning algorithm and $x \in \mathcal{X}$ is a fixed configuration of (tunable) hyper-parameters of the training algorithm. The objective is to find the hyper-parameter configuration that achieves the highest validation accuracy. An algorithm for this problem chooses, at each round $t$, an input (also called action or arm) $x_t \in \mathcal{X}$, and subsequently observes a function evaluation (also called reward) $y_t = f^*(x_t) + \varepsilon_t$, which is a noisy version of the function value at $x_t$. The action $x_t$ is chosen causally depending upon the history $\{x_1, y_1, \ldots, x_{t-1}, y_{t-1}\}$ of arms and reward sequences available before round $t$.

**Assumption 1 (Sub-Gaussian noise)** *The noise sequence $\{\varepsilon_t\}_{t\geq 1}$ is conditionally zero-mean, i.e., $\mathbb{E}\left[\varepsilon_t|\mathcal{F}_{t-1}\right] = 0$ and $\sigma$-sub-Gaussian for known $\sigma$ ,i.e.,*

$$\forall t \geq 1,\ \forall \lambda \in \mathbb{R},\ \mathbb{E}\left[\exp(\lambda \varepsilon_t)|\mathcal{F}_{t-1}\right] \leq \exp\left(\frac{\lambda^2\sigma^2}{2}\right)$$

*almost surely, where $\mathcal{F}_{t-1} := \sigma(x_1, y_1, \ldots, x_{t-1}, y_{t-1}, x_t)$ is the $\sigma$-field summarizing the information available just before $y_t$ is observed.*

This is a mild assumption on the noise (it holds, for instance, for distributions bounded in $[-\sigma, \sigma]$) and is standard in the literature [1, 26, 28].

**Regret:** The learner's goal is to maximize its (expected) cumulative reward $\sum_{t=1}^{t} f^*(x_t)$ over a time horizon $T$ (not necessarily known a priori) or, equivalently, minimize its cumulative *regret*

$$\mathcal{R}_T := \sum\nolimits_{t=1}^{T} \left(f^*(x^*) - f^*(x_t)\right),$$

where $x^* \in \operatorname{argmax}_{x\in\mathcal{X}} f(x)$ is a maximizer of $f$ (assuming the maximum is attained; not necessarily unique). A sublinear growth of $\mathcal{R}_T$ implies the time-average regret $\mathcal{R}_T/T \to 0$ as $T \to \infty$, implying the algorithm eventually chooses actions that attain function values close to the optimum most of the time.

### 2.1 Model Selection Objective

In the literature, it is assumed that $f^*$ belongs to a known class of functions $\mathcal{F}$. In this work, in contrast to the standard setting, we do not assume the knowledge of $\mathcal{F}$. Instead, we are given $M$ nested function classes $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \ldots \subset \mathcal{F}_M$. Among the nested classes $\mathcal{F}_1, ..\mathcal{F}_M$, the ones containing $f^*$ is denoted as *realizable* classes, and the ones not containing $f^*$ are dubbed as *non-realizable* classes. The

smallest such family where the unknown function $f^*$ lies is denoted by $\mathcal{F}_{m^*}$, where $m^* \in [M]$. However, we do not know the index $m^*$, and our goal is to propose adaptive algorithms such that the regret depends on the complexity of the function class $\mathcal{F}_{m^*}$. In order to achieve this, we need a separability condition on the nested models.

**Assumption 2 (Local Separability)** *There exist $\Delta > 0$ and $\eta > 0$ such that*

$$\inf_{f \in \mathcal{F}_{m^*-1}} \inf_{x_1 \neq x_2 : D^*(x_1,x_2) \leqslant \eta} |f(x_1) - f^*(x_2)| \geq \Delta,$$

*where*[4]*, $D^*(x_1, x_2) = |f^*(x_1) - f^*(x_2)|$.*

The above assumption[5] ensures that for action pairs $(x_1, x_2)$, where the obtained (expected) rewards are close (since it is generated by $f^*$), there is a gap between the true function $f^*$ and the ones belonging to the function classes not containing $f^*$ (i.e., the non-realizable function classes). Note that we do not require this separability to hold for all actions – just the ones which are indistinguishable from observing the rewards. Note that separability is needed for model selection since we neither assume any structural assumption on $f^*$, nor on the set $\mathcal{X}$.

We emphasize that separability is quite standard and assumptions of similar nature appear in a wide range of model selection problems, specially in the setting of contextual bandits [16, 19]. It is also quite standard in statistics, specifically in the area of clustering and latent variable modelling [5, 14, 34].

*Separability for Lipschitz $f^*$:* If the true function $f^*$ is 1-Lipschitz. In that setting, the separability assumption takes the following form: for $\Delta > 0$ and $\eta > 0$,

$$\inf_{f \in \mathcal{F}_{m^*-1}} \inf_{x_1 \neq x_2 : \|x_1 - x_2\| \leqslant \eta} |f(x_1) - f^*(x_2)| \geq \Delta$$

However, note that the above assumption is quite strong – any (random) arbitrary algorithm can perform model selection (with the knowledge of $\eta$ and $\Delta$)[6] in the following way: first choose action $x_1$. Using $\|x_1 - x_2\| \leqslant \eta$, choose $x_2$. Pick any function $f$ belonging to some class $\mathcal{F}_m$ in the nested family and evaluate $|f(x_1) - y_t(x_2)|$, which is a good proxy for $|f(x_1) - f^*(x_2)|$. The algorithm continues to pick different $f \in \mathcal{F}_m$. With the knowledge of $\Delta$, depending on how big $\mathcal{F}_m$ is, the algorithm would be able to identify whether $\mathcal{F}_m$ is realizable or not. Continuing it for all hypothesis classes, it would identify the correct class $\mathcal{F}_{m^*}$. Hence, for structured $f^*$, the problem of model selection with separation is not interesting and we do not consider that setup in this paper.

[4] Here the roles of $x_1$ and $x_2$ are interchangeable without loss of generality.

[5] We assume that the action set $\mathcal{X}$ is compact and continuous, and so such action pairs $(x_1, x_2)$ always exist, i.e., given any $x_1 \in \mathcal{X}$, an action $x_2$ such that $D^*(x_1, x_2) \leqslant \eta$ always exists.

[6] This can be found using standard trick like doubling.

*Separability for Linear $f^*$:* If $f^*$ is linear, the separability assumption is not necessary for model selection. In this setting, $f^*$ is parameterized by some properties of the parameter, such as sparsity and norm, denotes the nested function classes. [12, 15] addresses the linear bandit model selection problem without the separability assumption.

### 2.2 Algorithm: Adaptive Bandit Learning (ABL)

In this section, we provide a novel model selection algorithm (Algorithm 2) that, over multiple epochs, successively refine the estimate of the true model class $\mathcal{F}_{m^*}$ where the unknown function $f^*$ lies. At each epoch, we run a fresh instance of a base bandit algorithm for the estimated function class, which we call Bandit Learning. Note that our model selection algorithm works with any provable bandit learning algorithm, and is agnostic to the particular choice of such base algorithm. In what follows, we present a generic description of the base algorithm and then specialize to a special case.

*The Base Algorithm* Bandit Learning (Algorithm 1), in its general form, takes a function class $\mathcal{F}$ and a confidence level $\delta \in (0, 1]$ as its inputs. At each time $t$, it maintains a (high-probability) confidence set $\mathcal{C}_t(\mathcal{F}, \delta)$ for the unknown function $f^*$, and chooses the most optimistic action with respect to this confidence set,

$$x_t \in \operatorname*{argmax}_{x \in \mathcal{X}} \max_{f \in \mathcal{C}_t(\mathcal{F}, \delta)} f(x) \,. \tag{1}$$

The confidence set $\mathcal{C}_t(\mathcal{F}, \delta)$ is constructed using all the data $\{x_s, y_s\}_{s<t}$ gathered in the past. First, a regularized least square estimate of $f^*$ is computed as $\widehat{f}_t \in \operatorname{argmin}_{f \in \mathcal{F}} \mathcal{L}_{t-1}(f)$, where $\mathcal{L}_t(f) := \sum_{s=1}^{t} \left(y_s - f(x_s)\right)^2$ is the cumulative squared prediction error. The confidence set $\mathcal{C}_t(\mathcal{F}, \delta)$ is then defined as the set of all functions $f \in \mathcal{F}$ satisfying

$$\sum_{s=1}^{t-1} \left(f(x_s) - \widehat{f}_t(x_s)\right)^2 \leq \beta_t(\mathcal{F}, \delta) \,, \tag{2}$$

where $\beta_t(\mathcal{F}, \delta)$ is an appropriately chosen confidence parameter. We now specialize to the bandit learning algorithm of [26] by setting the confidence parameter

$$\beta_t(\mathcal{F}, \delta) := 8\sigma^2 \log\left(2\mathcal{N}\left(\mathcal{F}, 1/T, \|\cdot\|_\infty\right)/\delta\right) + 2\left(8 + \sqrt{8\sigma^2 \log\left(8t(t+1)/\delta\right)}\right),$$

where $\mathcal{N}(\mathcal{F}, \alpha, \|\cdot\|_\infty)$ is the $(\alpha, \|\cdot\|_\infty)$-covering number[7] of $\mathcal{F}$, one can ensure that $f^*$ lies in the confidence set $\mathcal{C}_t(\mathcal{F}, \delta)$ at all time instant $t \geq 1$ with probability at least $1 - \delta$. The theoretical guarantees presented in the paper are also for this particular choice of base algorithm.

[7] For any $\alpha > 0$, we call $\mathcal{F}^\alpha$ an $(\alpha, \|\cdot\|_\infty)$ cover of the function class $\mathcal{F}$ if for any $f \in \mathcal{F}$ there exists an $f'$ in $\mathcal{F}^\alpha$ such that $\|f' - f\|_\infty := \sup_{x \in \mathcal{X}} |f'(x) - f(x)| \leq \alpha$.

**Algorithm 1** Bandit Learning

**Input:** Function class $\mathcal{F}$, confidence level $\delta \in (0, 1]$
**for** time $t = 1, 2, 3, \ldots$ **do**
  Compute an estimate $\widehat{f}_t$ of $f^*$
  Construct confidence set $\mathcal{C}_t(\mathcal{F}, \delta)$ using (2)
  Choose an action $x_t$ using (1)
  Observe reward $y_t$
**end for**

**Algorithm 2** Adaptive Bandit Learning (`ABL`)

**Input:** Nested function classes $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \ldots \subset \mathcal{F}_M$, confidence level $\delta \in (0, 1]$, threshold $\gamma_i > 0$
**for** epochs $i = 1, 2 \ldots$ **do**
  **Model Selection:**
  Compute elapsed time $\tau_{i-1} = \sum_{j=1}^{i-1} t_j$
  **for** function classes $m = 1, 2 \ldots, M$ **do**
    Compute the minimum average squared prediction error using (3)
  **end for**
  Choose index $m^{(i)} = \min\{m \in [M] : T_m^{(i)} \leq \gamma_i\}$
  **Model Learning:**
  Set epoch length $t_i = 2^i$, confidence level $\delta_i = \delta/2^i$
  Run Bandit Learning (Algorithm 1) over a time horizon $t_i$ with function class $\mathcal{F}_{m^{(i)}}$ and confidence level $\delta_i$ as its inputs
**end for**

*Our Approach–Adaptive Bandit Learning (`ABL`):* The description of our model selection algorithm is given in Algorithm 2. We consider doubling epochs – at each epoch $i \geq 1$, the base algorithm is run over time horizon $t_i = 2^i$. At the beginning of $i$-th epoch, using all the data of the previous epochs, we employ a model selection module as follows. First, we compute, for each class $\mathcal{F}_m$, the minimum average squared prediction error (via an offline regression oracle)

$$T_m^{(i)} = \min_{f \in \mathcal{F}_m} \frac{1}{\tau_{i-1}} \sum_{s=1}^{\tau_{i-1}} \left(y_s - f(x_s)\right)^2 \ , \tag{3}$$

where $\tau_{i-1} := \sum_{j=1}^{i-1} t_j$ denotes the total time elapsed before epoch $i$. Finally, we compare $T_m^{(i)}$ to a pre-calculated threshold $\gamma$, and pick the function class for which $T_m^{(i)}$ falls below such threshold (with smallest $m$, see Algorithm 2). After selecting the function class, we run the base algorithm for this class with confidence level $\delta_i = \delta/2^i$. We call the complete procedure Adaptive Bandit Learning (`ABL`).

### 2.3 Performance Guarantee of ABL

We now provide model selection and regret guarantees of ABL (Algorithm 2), when the base algorithm is chosen as [26]. Though the results to be presented in this section are quite general, they do not apply to any arbitrary function classes. In what follows, we will make the following boundedness assumption.

**Assumption 3 (Bounded functions)** *We assume that* $f(x) \in [0,1]\ \forall\ x \in \mathcal{X}$ *and* $f \in \mathcal{F}_m$ *(*$\forall\ m \in [M]$*).*[8]

It is worth noting that this same assumption is also required in the standard setting, i.e., when the true model class is known ($\mathcal{F}_{m^*} = \mathcal{F}$).

We denote by $\log \mathcal{N}(\mathcal{F}_m) = \log\left(\mathcal{N}(\mathcal{F}_m, 1/T, \|\cdot\|_\infty)\right)$ the metric entropy (with scale $1/T$) of the class $\mathcal{F}_m$. We have the following guarantee for ABL.

**Lemma 1 (Model selection of ABL).** *Fix a* $\delta \in (0,1]$ *and* $\lambda > 0$*. Suppose, Assumptions 1, 2 and 3 hold and we set the threshold* $\gamma_i = T_M^{(i)} + C_1$*, for a sufficiently small constant* $C_1$*. Then, with probability at least* $1 - O(M\delta)$*, ABL identifies the correct model class* $\mathcal{F}_{m^*}$ *from epoch* $i \geq i^*$ *when the time elapsed before epoch* $i^*$ *satisfies*

$$\tau_{i^*-1} \geq C\sigma^4(\log T)\max\left\{\frac{\log(1/\delta)}{(\frac{\Delta^2}{2} - 4\eta)^2}, \log\left(\frac{\mathcal{N}(\mathcal{F}_M)}{\delta}\right)\right\},$$

*provided* $\Delta \geqslant 2\sqrt{2\eta}$*, where* $C > 1$ *is a sufficiently large universal constant.*

*Remark 1 (Dependence on the biggest class).* Note that we choose a threshold that depends on the epoch number and the test statistic of the biggest class. Here we crucially exploit the fact that the biggest class always contains the true model class and use this to design the threshold.

We characterize the complexity of each function class $\mathcal{F}_m$ by its *eluder dimension*, first introduced by [26] in the standard setting.

**Definition 1 (Eluder dimension).** *The* $\varepsilon$*-eluder dimension* $\dim_{\mathcal{E}}(\mathcal{F}_m, \varepsilon)$ *of a function class* $\mathcal{F}$ *is the length of the longest sequence* $\{x_i\}_{i=1}^n \subseteq \mathcal{X}$ *of input points such that for some* $\varepsilon' \geq \varepsilon$ *and for each* $i \in \{2, \ldots, n\}$*,*

$$\sup_{f_1, f_2 \in \mathcal{F}}\left\{(f_1 - f_2)(x_i) \;\middle|\; \sqrt{\sum_{j=1}^{i-1}(f_1 - f_2)^2(x_i)} \leq \varepsilon'\right\} > \varepsilon'\ .$$

Define $\mathcal{F}^* = \mathcal{F}_{m^*}$. Denote by $d_{\mathcal{E}}(\mathcal{F}^*) = \dim_{\mathcal{E}}(\mathcal{F}^*, 1/T)$, the $(1/T)$-eluder dimension of the (realizable) function class $\mathcal{F}^*$, where $T$ is the time horizon. Then, armed with Lemma 1, we obtain the following regret bound for ABL.

[8] We can extent the range to $[0, c]$ without loss of generality.

**Theorem 1 (Cumulative regret of ABL).** *Suppose the condition of Lemma 1 holds. Then, for any $\delta \in (0,1]$, the regret of ABL for horizon $T$ is*

$$\mathcal{R}_T \leq \mathcal{O}\left(\sigma^4(\log T)\max\left\{\frac{\log(1/\delta)}{(\frac{\Delta^2}{2}-4\eta)^2}, \log\left(\frac{\mathcal{N}(\mathcal{F}_M)}{\delta}\right)\right\}\right) + \mathcal{O}\left(d_{\mathcal{E}}(\mathcal{F}^*)\log T + c\sqrt{T d_{\mathcal{E}}(\mathcal{F}^*)\log(\mathcal{N}(\mathcal{F}^*)/\delta)\log^2(T/\delta)}\right),$$

*with probability at least*[9] $1 - O(M\delta)$.

*Remark 2 (Cost of model selection).* We retain the regret bound of [26] in the standard setting, and the first term in the regret bound captures the cost of model selection – the cost suffered before accumulating enough samples to infer the correct model class (with high probability). It has weak (logarithmic) dependence on horizon $T$ and hence considered as a minor term, in the setting where $T$ is large. Hence, model selection is essentially *free* upto log factors. Let us now have a close look at this term. It depends on the metric entropy of the biggest model class $\mathcal{F}_M$. This stems from the fact that the thresholds $\{\gamma_i\}_{i\geq 1}$ depend on the test statistic of $\mathcal{F}_M$ (see Remark 1). We believe that, without additional assumptions, one can't get rid of this (minor) dependence on the complexity of the biggest class.

The second term is the major one ($\sqrt{T}$ dependence on total number of steps), which essentially is the cost of learning the true kernel $f^*$. Since in this phase, we basically run the base algorithm for the correct model class, our regret guarantee matches to that of an oracle with the apriori knowledge of the correct class. Note that if we simply run a non model-adaptive algorithm for this problem, the regret would be $\widetilde{\mathcal{O}}(H\sqrt{T d_{\mathcal{E}}(\mathcal{F}_M)\log \mathcal{N}(\mathcal{F}_M)})$, where $d_{\mathcal{E}}(\mathcal{F}_M)$ denotes the eluder dimension of the largest model class $\mathcal{F}_M$. In contrast, by successively testing and thresholding, our algorithm adapts to the complexity of the smallest function class containing the true model class.

*Remark 3 (Requires no knowledge of $(\Delta, \eta)$).* Our algorithm ABL doesn't require the knowledge of $\Delta$ and $\eta$. Rather, it automatically adapts to these parameters, and the dependence is reflected in the regret expression. The separation $\Delta$ implies how complex the job of model selection is. If the separation is small, it is difficult for ABL to separate out the model classes. Hence, it requires additional exploration, and as a result the regret increases. Another interesting fact of Theorem 1 is that it does not require any minimum separation across model classes. This is in sharp contrast with existing results in statistics (see, e.g. [5,34]). Even if $\Delta$ is quite small, Theorem 1 gives a model selection guarantee. Now, the cost of separation appears anyways in the minor term, and hence in the long run, it does not effect the overall performance of the algorithm.

[9] One can choose $\delta = 1/\text{poly}(M)$ to obtain a high-probability bound which only adds an extra $\log M$ factor.

## 3 Model Selection in Markov Decision Processes

An (episodic) MDP is denoted by $\mathcal{M}(\mathcal{S}, \mathcal{A}, H, P^*, r)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space (both possibly infinite), $H$ is the length of each episode, $P^* : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is an (unknown) transition kernel (a function mapping state-action pairs to signed distribution over the state space) and $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is a (known) reward function. In episodic MDPs, a (deterministic) policy $\pi$ is given by a collection of $H$ functions $(\pi_1, \ldots, \pi_H)$, where each $\pi_h : \mathcal{S} \to \mathcal{A}$ maps a state $s$ to an action $a$. In each episode, an initial state $s_1$ is first picked by the environment (assumed to be fixed and history independent). Then, at each step $h \in [H]$, the agent observes the state $s_h$, picks an action $a_h$ according to $\pi_h$, receives a reward $r(s_h, a_h)$, and then transitions to the next state $s_{h+1}$, which is drawn from the conditional distribution $P^*(\cdot|s_h, a_h)$. The episode ends when the terminal state $s_{H+1}$ is reached. For each state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ and step $h \in [H]$, we define action values $Q_h^\pi(s, a)$ and and state values $V_h^\pi(s)$ corresponding to a policy $\pi$ as

$$Q_h^\pi(s,a) = r(s,a) + \mathbb{E}\left[\sum\nolimits_{h'=h+1}^{H} r(s_{h'}, \pi_{h'}(s_{h'}))|s_h = s, a_h = a\right], \quad V_h^\pi(s) = Q_h^\pi\big(s, \pi_h(s)\big) \ ,$$

where the expectation is with respect to the randomness of the transition distribution $P^*$. It is not hard to see that $Q_h^\pi$ and $V_h^\pi$ satisfy the Bellman equations:

$$Q_h^\pi(s,a) = r(s,a) + (P^* V_{h+1}^\pi)(s,a) \,, \ \ \forall h \in [H], \quad \text{with } V_{H+1}^\pi(s) = 0 \text{ for all } s \in \mathcal{S}.$$

A policy $\pi^*$ is said to be optimal if it maximizes the value for all states $s$ and step $h$ simultaneously, and the corresponding optimal value function is denoted by $V_h^*(s) = \sup_\pi V_h^\pi(s)$ for all $h \in [H]$, where the supremum is over all (non-stationary) policies. The agent interacts with the environment for $K$ episodes to learn the unknown transition kernel $P^*$ and thus, in turn, the optimal policy $\pi^*$. At each episode $k \geq 1$, the agent chooses a policy $\pi^k := (\pi_1^k, \ldots, \pi_H^k)$ and a trajectory $(s_h^k, a_h^k, r(s_h^k, a_h^k), s_{h+1}^k)_{h \in [H]}$ is generated. The performance of the learning agent is measured by the cumulative (pseudo) regret accumulated over $K$ episodes, defined as

$$\mathcal{R}(T) := \sum\nolimits_{k=1}^{K} \left[V_1^*(s_1^k) - V_1^{\pi^k}(s_1^k)\right],$$

where $T = KH$ is total steps in $K$ episodes.

In this work, we consider general MDPs without any structural assumption on the unknown transition kernel $P^*$. In the standard setting [4], it is assumed that $P^*$ belongs to a known family of transition models $\mathcal{P}$. Here, in contrast to the standard setting, we do not have the knowledge of $\mathcal{P}$. Instead, we are given $M$ nested families of transition kernels $\mathcal{P}_1 \subset \mathcal{P}_2 \subset \ldots \subset \mathcal{P}_M$. The smallest such family where the true transition kernel $P^*$ lies is denoted by $\mathcal{P}_{m^*}$, where $m^* \in [M]$. However, we do not know the index $m^*$, and our goal is to propose adaptive algorithms such that the regret depends on the complexity of the family $\mathcal{P}_{m^*}$. We assume a similar separability condition on these nested model classes.

**Assumption 4 (Local Separability)** *There exist constants $\Delta > 0$ and $\eta > 0$ such that for any function $V : \mathcal{S} \to \mathbb{R}$,*

$$\inf_{P \in \mathcal{P}_{m^*-1}} \inf_{D*((s_1,a_1),(s_2,a_2)) \leqslant \eta} |PV(s_1,a_1) - P^*V(s_2,a_2)| \geq \Delta,$$

*where $(s_1,a_1) \neq (s_2,a_2)$ and $D^*((s_1,a_1),(s_2,a_2)) = |P^*V(s_1,a_1) - P^*V(s_2,a_2)|$.*

This assumption ensures that expected values under the true model is well-separated from those under models from non-realizable classes for two distinct state-action pairs for which values are close under true model. Once again, we need state and action spaces to be compact and continuous to guarantee such pairs always exist. Note that the assumption might appear to break down for any constant function $V$. However, we will be invoking this assumption with the value functions computed by the learning algorithm (see (4)). For reward functions that *vary sufficiently* across states and actions, and transition kernels that admit densities, the chance of getting hit by constant value functions is admissibly low. In case the rewards are constant, every policy would anyway incur zero regret rendering the learning problem trivial. The value functions appear in the separability assumption in the first place since we are interested in minimizing the regret. Instead, if one cares only about learning the true model, then separability of transition kernels under some suitable notion of distance (e.g., the KL-divergence) might suffice. Note that in [16,19], the regret is defined in terms of the regression function and hence the separability is assumed on the regression function itself. Model selection without separability is kept as an interesting future work.

### 3.1 Algorithm: Adaptive Reinforcement Learning (`ARL`)

In this section, we provide a novel model selection algorithm `ARL` (Algorithm 2) that use successive refinements over epochs. We use `UCRL-VTR` algorithm of [4] as our base algorithm, and add a model selection module at the beginning of each epoch. In other words, over multiple epochs, we successively refine our estimates of the proper model class where the true transition kernel $P^*$ lies.

*The Base Algorithm:* `UCRL-VTR`, in its general form, takes a family of transition models $\mathcal{P}$ and a confidence level $\delta \in (0,1]$ as its input. At each episode $k$, it maintains a (high-probability) confidence set $\mathcal{B}_{k-1} \subset \mathcal{P}$ for the unknown model $P^*$ and use it for optimistic planning. First, it finds the transition kernel $P_k = \text{argmax}_{P \in \mathcal{B}_{k-1}} V^*_{P,1}(s^k_1)$, where $V^*_{P,h}$ denote the optimal value function of an MDP with transition kernel $P$ at step $h$. `UCRL-VTR` then computes, at each step $h$, the optimal value function $V^k_h := V^*_{P_k,h}$ under the kernel $P_k$ using dynamic programming. Specifically, starting with $V^k_{H+1}(s,a) = 0$ for all pairs $(s,a)$, it defines for all steps $h = H$ down to 1,

$$Q^k_h(s,a) = r(s,a) + (P_k V^k_{h+1})(s,a), \quad V^k_h(s) = \max_{a \in \mathcal{A}} Q^k_h(s,a). \tag{4}$$

Then, at each step $h$, UCRL-VTR takes the action that maximizes the $Q$-function estimate, i,e. it chooses $a_h^k = \text{argmax}_{a\in\mathcal{A}} Q_h^k(s_h^k, a)$. Now, the confidence set is updated using all the data gathered in the episode. First, UCRL-VTR computes an estimate of $P^*$ by employing a non-linear value-targeted regression model with data $\left(s_h^j, a_h^j, V_{h+1}^j(s_{h+1}^j)\right)_{j\in[k],h\in[H]}$. Note that $\mathbb{E}[V_{h+1}^k(s_{h+1}^k)|\mathcal{G}_{h-1}^k] = (P^*V_{h+1}^k)(s_h^k, a_h^k)$, where $\mathcal{G}_{h-1}^k$ denotes the $\sigma$-field summarizing the information available just before $s_{h+1}^k$ is observed. This naturally leads to the estimate $\widehat{P}_k = \text{argmin}_{P\in\mathcal{P}} \mathcal{L}_k(P)$, where

$$\mathcal{L}_k(P) := \sum_{j=1}^{k} \sum_{h=1}^{H} \left(V_{h+1}^j(s_{h+1}^j) - (PV_{h+1}^j)(s_h^j, a_h^j)\right)^2. \tag{5}$$

The confidence set $\mathcal{B}_k$ is then updated by enumerating the set of all transition kernels $P \in \mathcal{P}$ satisfying $\sum_{j=1}^{k}\sum_{h=1}^{H}\left((PV_{h+1}^j)(s_h^j, a_h^j) - (\widehat{P}_k V_{h+1}^j)(s_h^j, a_h^j)\right)^2 \leq \beta_k(\delta)$ with the confidence width being defined as $\beta_k(\delta) := 8H^2 \log\left(\frac{2\mathcal{N}(\mathcal{P}, \frac{1}{kH}, \|\cdot\|_{\infty,1})}{\delta}\right) + 4H^2\left(2 + \sqrt{2\log\left(\frac{4kH(kH+1)}{\delta}\right)}\right)$, where $\mathcal{N}(\mathcal{P}, \cdot, \cdot)$ denotes the covering number of the family $\mathcal{P}$. [10] Then, one can show that $P^*$ lies in the confidence set $\mathcal{B}_k$ in all episodes $k$ with probability at least $1-\delta$. Here, we consider a slight different expression of $\beta_k(\delta)$ as compared to [4], but the proof essentially follows the same technique. Please refer to Appendix B for further details.

*Our Approach:* We consider doubling epochs - at each epoch $i \geq 1$, UCRL-VTR is run for $k_i = 2^i$ episodes. At the beginning of $i$-th epoch, using all the data of previous epochs, we add a model selection module as follows. First, we compute, for each family $\mathcal{P}_m$, the transition kernel $\widehat{P}_m^{(i)}$, that minimizes the empirical loss $\mathcal{L}_{\tau_{i-1}}(P)$ over all $P \in \mathcal{P}_m$ (see (5)), where $\tau_{i-1} := \sum_{j=1}^{k-1} k_j$ denotes the total number of episodes completed before epoch $i$. Next, we compute the average empirical loss $T_m^{(i)} := \frac{1}{\tau_{i-1}H}\mathcal{L}_{\tau_{i-1}}(\widehat{P}_m^{(i)})$ for the model $\widehat{P}_m^{(i)}$. Finally, we compare $T_m^{(i)}$ to a pre-calculated threshold $\gamma_i$, and pick the transition family for which $T_m^{(i)}$ falls below such threshold (with smallest $m$, see Algorithm 3). After selecting the family, we run UCRL-VTR for this family with confidence level $\delta_i = \frac{\delta}{2^i}$, where $\delta \in (0, 1]$ is a parameter of the algorithm.

### 3.2 Performance Guarantee of ARL

First, we present our main result which states that the model selection procedure of ARL (Algorithm 3) succeeds with high probability after a certain number of epochs. To this end, we denote by $\log\mathcal{N}(\mathcal{P}_m) = \log(\mathcal{N}(\mathcal{P}_m, 1/T, \|\cdot\|_{\infty,1}))$ the metric entropy (with scale $1/T$) of the family $\mathcal{P}_m$. We also use the shorthand notation $\mathcal{P}^* = \mathcal{P}_{m^*}$.

[10] For any $\alpha > 0$, $\mathcal{P}^\alpha$ is an $(\alpha, \|\cdot\|_{\infty,1})$ cover of $\mathcal{P}$ if for any $P \in \mathcal{P}$ there exists an $P'$ in $\mathcal{P}^\alpha$ such that $\|P' - P\|_{\infty,1} := \sup_{s,a}\int_{\mathcal{S}} |P'(s'|s,a) - P(s'|s,a)|ds' \leq \alpha$.

**Algorithm 3** Adaptive Reinforcement Learning – `ARL`

**Input:** Parameter $\delta$, function classes $\mathcal{P}_1 \subset \mathcal{P}_2 \subset \ldots \subset \mathcal{P}_M$, thresholds $\{\gamma_i\}_{i\geq 1}$
**for** epochs $i = 1, 2 \ldots$ **do**
  Set $\tau_{i-1} = \sum_{j=1}^{i-1} k_j$
  **for** function classes $m = 1, 2 \ldots, M$ **do**
    Compute $\widehat{P}_m^{(i)} = \operatorname{argmin}_{P\in\mathcal{P}_m} \sum_{k=1}^{\tau_{i-1}} \sum_{h=1}^{H} \big(V_{h+1}^k(s_{h+1}^k) - (PV_{h+1}^k)(s_h^k, a_h^k)\big)^2$
    Compute $T_m^{(i)} = \frac{1}{\tau_{i-1}H} \sum_{k=1}^{\tau_{i-1}} \sum_{h=1}^{H} \Big(V_{h+1}^k(s_{h+1}^k) - (\widehat{P}_m^{(i)} V_{h+1}^k)(s_h^k, a_h^k)\Big)^2$
  **end for**
  Set $m^{(i)} = \min\{m \in [M] : T_m^{(i)} \leq \gamma_i\}$, $k_i = 2^i$ and $\delta_i = \delta/2^i$
  Run `UCRL-VTR` for the family $\mathcal{P}_{m^{(i)}}$ for $k_i$ episodes with confidence level $\delta_i$
**end for**

**Lemma 2 (Model selection of `ARL`).** *Fix a $\delta \in (0, 1]$ and suppose Assumption 2 holds. Suppose the thresholds are set as $\gamma_i = T_M^{(i)} + C_2$, for some sufficiently small constant $C_2$. Then, with probability at least $1 - O(M\delta)$, `ARL` identifies the correct model class $\mathcal{P}_{m^*}$ from epoch $i \geqslant i^*$, where epoch length of $i^*$ satisfies*

$$2^{i^*} \geqslant C' \log K \max\left\{ \frac{H^3}{(\frac{1}{2}\Delta^2 - 2H\eta)^2} \log(2/\delta), 4H \log\left(\frac{\mathcal{N}(\mathcal{P}_M)}{\delta}\right)\right\},$$

*provided $\Delta \geqslant 2\sqrt{H\eta}$, for a sufficiently large universal constant $C' > 1$.*

*Regret Bound:* In order to present our regret bound, we define, for each model model class $\mathcal{P}_m$, a collection of functions $\mathcal{M}_m := \{f : \mathcal{S} \times \mathcal{A} \times \mathcal{V}_m \to \mathbb{R}\}$ such that any $f \in \mathcal{M}_m$ satisfies $f(s, a, V) = (PV)(s, a)$ for some $P \in \mathcal{P}_m$ and $V \in \mathcal{V}_m$, where $\mathcal{V}_m := \{V_{P,h}^* : P \in \mathcal{P}_m, h \in [H]\}$ denotes the set of optimal value functions under the transition family $\mathcal{P}_m$. By one-to-one correspondence, we have $\mathcal{M}_1 \subset \mathcal{M}_2 \subset \ldots \subset \mathcal{M}_M$, and the complexities of these function classes determine the learning complexity of the RL problem under consideration. We characterize the complexity of each function class $\mathcal{M}_m$ by its *eluder dimension*, which is defined similarly as Definition 1. (We take domain of function class $\mathcal{M}_m$ to be $\mathcal{S} \times \mathcal{A} \times \mathcal{V}_m$.)

We define $\mathcal{M}^* = \mathcal{M}_{m^*}$, and denote by $d_{\mathcal{E}}(\mathcal{M}^*) = \dim_{\mathcal{E}}(\mathcal{M}^*, 1/T)$, the $(1/T)$-eluder dimension of the (realizable) function class $\mathcal{M}^*$, where $T$ is the time horizon. Then, armed with Lemma 2, we obtain the following regret bound.

**Theorem 2 (Cumulative regret of `ARL`).** *Suppose the conditions of Lemma 2 hold. Then, for any $\delta \in (0, 1]$, running `ARL` for $K$ episodes yields a regret bound*

$$\mathcal{R}(T) = \mathcal{O}\left(\log K \max\left\{\frac{H^4 \log(1/\delta)}{(\frac{\Delta^2}{2} - 2H\eta)^2}, H^2 \log\left(\frac{\mathcal{N}(\mathcal{P}_M)}{\delta}\right)\right\}\right)$$
$$+ \mathcal{O}\left(H^2 d_{\mathcal{E}}(\mathcal{M}^*) \log K + H\sqrt{T d_{\mathcal{E}}(\mathcal{M}^*) \log(\mathcal{N}(\mathcal{P}^*)/\delta) \log K \log(T/\delta)}\right).$$

*with probability at least $1 - O(M\delta)$.*

Similar to Theorem 1, the first term in the regret bound captures the cost of model selection, having weak (logarithmic) dependence on the number of episodes $K$ and hence considered as a minor term, in the setting where $K$ is large. Hence, model selection is essentially *free* upto log factors. The second term is the major one ($\sqrt{T}$ dependence on total number of steps), which essentially is the cost of learning the true kernel $P^*$. Since in this phase, we basically run `UCRL-VTR` for the correct model class, our regret guarantee matches to that of an oracle with the apriori knowledge of the correct class. `ARL` doesnot require the knowledge of $(\Delta, \eta)$ and it adapts to the complexity of the problem.

## 4 Conclusion

We address the problem of model selection for MAB and MDP and propose algorithms that obtains regret similar to an oracle who knows the true model class apriori. Our algorithms leverage the separability conditions crucially, and removing them is kept as a future work.

## Acknowledgements

We thank anonymous reviewers for their useful comments. Moreover, we would like to thank Prof. Kannan Ramchandran (EECS, UC Berkeley) for insightful discussions regarding the topic of model selection. SRC is grateful to a CISE postdoctoral fellowship of Boston University.

## References

1. Abbasi-Yadkori, Y., Pál, D., Szepesvári, C.: Improved algorithms for linear stochastic bandits. In: Advances in Neural Information Processing Systems. pp. 2312–2320 (2011)
2. Agarwal, A., Luo, H., Neyshabur, B., Schapire, R.E.: Corralling a band of bandit algorithms. In: Conference on Learning Theory. pp. 12–38. PMLR (2017)
3. Arora, R., Marinov, T.V., Mohri, M.: Corralling stochastic bandit algorithms. In: International Conference on Artificial Intelligence and Statistics. pp. 2116–2124. PMLR (2021)
4. Ayoub, A., Jia, Z., Szepesvari, C., Wang, M., Yang, L.F.: Model-based reinforcement learning with value-targeted regression. arXiv preprint arXiv:2006.01107 (2020)
5. Balakrishnan, S., Wainwright, M.J., Yu, B., et al.: Statistical guarantees for the em algorithm: From population to sample-based analysis. Annals of Statistics **45**(1), 77–120 (2017)
6. Besbes, O., Zeevi, A.: Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. Operations Research **57**(6), 1407–1420 (2009)
7. Cesa-Bianchi, N., Lugosi, G.: Prediction, learning, and games. Cambridge University Press (2006)

8. Chang, F., Lai, T.L.: Optimal stopping and dynamic allocation. Advances in Applied Probability **19**(4), 829–853 (1987), http://www.jstor.org/stable/1427104
9. Chatterji, N.S., Muthukumar, V., Bartlett, P.L.: Osom: A simultaneously optimal algorithm for multi-armed and linear contextual bandits. arXiv preprint arXiv:1905.10040 (2019)
10. Chowdhury, S.R., Gopalan, A.: Online learning in kernelized markov decision processes. In: The 22nd International Conference on Artificial Intelligence and Statistics. pp. 3197–3205 (2019)
11. Even-Dar, E., Mannor, S., Mansour, Y.: Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. Journal of Machine Learning Research **7**(39), 1079–1105 (2006), http://jmlr.org/papers/v7/evendar06a.html
12. Foster, D.J., Krishnamurthy, A., Luo, H.: Model selection for contextual bandits. In: Advances in Neural Information Processing Systems. pp. 14714–14725 (2019)
13. Ghosh, A., Chowdhury, S.R., Gopalan, A.: Misspecified linear bandits. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 31 (2017)
14. Ghosh, A., Pananjady, A., Guntuboyina, A., Ramchandran, K.: Max-affine regression: Provable, tractable, and near-optimal statistical estimation. arXiv preprint arXiv:1906.09255 (2019)
15. Ghosh, A., Sankararaman, A., Kannan, R.: Problem-complexity adaptive model selection for stochastic linear bandits. In: International Conference on Artificial Intelligence and Statistics. pp. 1396–1404. PMLR (2021)
16. Ghosh, A., Sankararaman, A., Ramchandran, K.: Model selection for generic contextual bandits. arXiv preprint arXiv:2107.03455 (2021)
17. Jin, C., Yang, Z., Wang, Z., Jordan, M.I.: Provably efficient reinforcement learning with linear function approximation. arXiv preprint arXiv:1907.05388 (2019)
18. Kakade, S., Krishnamurthy, A., Lowrey, K., Ohnishi, M., Sun, W.: Information theoretic regret bounds for online nonlinear control. arXiv preprint arXiv:2006.12466 (2020)
19. Krishnamurthy, S.K., Athey, S.: Optimal model selection in contextual bandits with many classes via offline oracles. arXiv preprint arXiv:2106.06483 (2021)
20. Lee, J.N., Pacchiano, A., Muthukumar, V., Kong, W., Brunskill, E.: Online model selection for reinforcement learning with function approximation. CoRR **abs/2011.09750** (2020), https://arxiv.org/abs/2011.09750
21. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)
22. Osband, I., Van Roy, B.: Model-based reinforcement learning and the eluder dimension. In: Advances in Neural Information Processing Systems 27 (NIPS). pp. 1466–1474 (2014)
23. Pacchiano, A., Dann, C., Gentile, C., Bartlett, P.: Regret bound balancing and elimination for model selection in bandits and rl. arXiv preprint arXiv:2012.13045 (2020)
24. Pacchiano, A., Phan, M., Abbasi Yadkori, Y., Rao, A., Zimmert, J., Lattimore, T., Szepesvari, C.: Model selection in contextual stochastic bandit problems. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 10328–10337. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/751d51528afe5e6f7fe95dece4ed32ba-Paper.pdf

25. Puterman, M.L.: Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons (2014)
26. Russo, D., Van Roy, B.: Eluder dimension and the sample complexity of optimistic exploration. In: Advances in Neural Information Processing Systems. pp. 2256–2264 (2013)
27. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al.: Mastering the game of go without human knowledge. nature **550**(7676), 354–359 (2017)
28. Srinivas, N., Krause, A., Kakade, S.M., Seeger, M.W.: Information-theoretic regret bounds for gaussian process optimization in the bandit setting. IEEE Transactions on Information Theory **58**(5), 3250–3265 (2012)
29. Wang, R., Salakhutdinov, R., Yang, L.F.: Provably efficient reinforcement learning with general value function approximation. arXiv preprint arXiv:2005.10804 (2020)
30. Wang, Y., Wang, R., Du, S.S., Krishnamurthy, A.: Optimism in reinforcement learning with generalized linear function approximation. arXiv preprint arXiv:1912.04136 (2019)
31. Williams, G., Aldrich, A., Theodorou, E.A.: Model predictive path integral control: From theory to parallel computation. Journal of Guidance, Control, and Dynamics **40**(2), 344–357 (2017)
32. Yang, L.F., Wang, M.: Reinforcement leaning in feature space: Matrix bandit, kernels, and regret bound. arXiv preprint arXiv:1905.10389 (2019)
33. Yang, Z., Jin, C., Wang, Z., Wang, M., Jordan, M.: Provably efficient reinforcement learning with kernel and neural function approximations. Advances in Neural Information Processing Systems **33** (2020)
34. Yi, X., Caramanis, C., Sanghavi, S.: Solving a mixture of many random linear equations by tensor decomposition and alternating minimization. CoRR **abs/1608.05749** (2016), http://arxiv.org/abs/1608.05749
35. Zanette, A., Brandfonbrener, D., Brunskill, E., Pirotta, M., Lazaric, A.: Frequentist regret bounds for randomized least-squares value iteration. In: International Conference on Artificial Intelligence and Statistics. pp. 1954–1964 (2020)