Abstraction Selection in Model-Based Reinforcement Learning

Nan Jiang NANJIANG@UMICH.EDU
Alex Kulesza KULESZA@UMICH.EDU
Satinder Singh BAVEJA@UMICH.EDU
Computer Science & Engineering, University of Michigan

Abstract

State abstractions are often used to reduce the complexity of model-based reinforcement learning when only limited quantities of data are available. However, choosing the appropriate level of abstraction is an important problem in practice. Existing approaches have theoretical guarantees only under strong assumptions on the domain or asymptotically large amounts of data, but in this paper we propose a simple algorithm based on statistical hypothesis testing that comes with a finite-sample guarantee under assumptions on candidate abstractions. Our algorithm trades off the low approximation error of finer abstractions against the low estimation error of coarser abstractions, resulting in a loss bound that depends only on the quality of the *best* available abstraction and is polynomial in planning horizon.

1. Introduction

In this paper, we advance the theoretical understanding of a fundamentally important setting in reinforcement learning (RL): sequential decision-making problems with large state spaces but only limited amounts of data and no pre-existing model. This is, of course, the typical setting for many RL applications, and a number of algorithms that exploit some form of compact function approximation either to learn a model or to directly learn value functions or policies have been applied successfully across domains from control, robotics, resource allocation, and others. Examples of such methods include value function approximation (Sutton & Barto, 1998), policy-gradient methods (Sutton et al., 1999), kernel RL and related non-parametric dynamic programming algorithms (Ormoneit & Sen, 2002; Lever et al., 2012), and pre-processing with state abstraction/aggregation followed by standard RL algorithms (Li et al., 2006).

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

However, state-of-the-art theoretical analysis in this area mostly either (1) makes structural assumptions about the domain (e.g., linear dynamics (Parr et al., 2008)) to allow an RL algorithm using a fixed and finite-capacity function approximator to guarantee bounded loss as the size of the dataset grows to infinity, or (2) makes smoothness assumptions about the domain (e.g., Ormoneit & Sen, 2002) but guarantees zero loss only when both the function approximation capacity and the dataset size go to infinity. In contrast, we are interested in analyzing the more realistic case where no assumptions about the domain can be made—other than that it can be described by a Markov decision process (MDP)—and the dataset is finite.

In particular, we consider a scenario in which a domain expert offers a set of possible state abstractions for a given domain. We assume that these abstractions are finite aggregations of states; for instance, the expert may provide discrete-valued state features, implicitly defining an abstraction that aggregates states with identical feature values. Given a finite amount of data, our task is to discover which abstraction to use for computing a policy from the data. If the dataset is large, we should prefer finer abstractions that are faithful to the domain (those with low *approximation* error), but for smaller datasets, coarse, lossy abstractions may be preferable because they simplify learning (low *estimation* error).

To simplify our analysis, we assume the dataset is fixed in advance. To remove the choice of RL algorithm from our analysis, we assume *certainty equivalence*, i.e., we assume that the agent behaves optimally with respect to the maximum-likelihood model estimated from the data under the chosen abstraction. When the quality of the abstraction is known, the theory of approximate homomorphisms in MDPs bounds the loss of the certainty equivalence policy (Even-Dar & Mansour, 2003; Ravindran, 2004). However, here the quality of the abstractions is unknown, and must itself be estimated from data. Existing theoretical results in this setting either have exponential dependence on the effective planning horizon (Mandel et al., 2014), or apply to the online setting and depend on the total size of all abstract state spaces under consideration (Ortner et al.,

2014). For our purposes the latter result is no better than simply always choosing the finest abstraction.

Initially, we consider choosing between two abstractions, one of which is a refinement of the other (e.g., the finer abstraction uses a superset of the features of the coarser abstraction). We propose a simple algorithm, and prove a theoretical guarantee that only depends on the *better* abstraction and is *polynomial* in effective planning horizon. Then, we show how to extend our analysis to an arbitrary set of abstractions that are successive refinements.

The algorithm we present and analyze is similar to existing algorithms that aggregate/split states via hypothesis testing with various state aliasing criteria (Jong & Stone, 2005; Dinculescu & Precup, 2010; Talvitie & Singh, 2011; Hallak et al., 2013). However, our analysis provides the first finite-sample guarantee theoretically justifying this family of methods. Previous theoretical work has assumed that at least one of the candidate abstractions is perfect and will be discovered asymptotically in the limit of data (e.g., Hallak et al., 2013, Section 5). However, abstractions are usually approximate in practice, and we need abstractions in the first place primarily because the data is *insufficient*. Asymptotic analyses offer little guidance for balancing approximation error and estimation error in this setting. Our analysis shows that a carefully designed hypothesis test can balance this finite-sample tradeoff even when none of the abstractions are perfect, and works almost as well as if the abstraction qualities were known in advance.

The rest of the paper is organized as follows. Section 2 introduces preliminaries and defines the abstraction selection problem. Section 3 develops a bound on the loss of a single abstraction, setting up the approximation and estimation error trade-off. Section 4 proposes and analyzes our algorithm. Section 5 reviews other approaches to the abstraction selection problem, and finally we conclude in Section 6.

2. Preliminaries

2.1. Markov Decision Processes (MDPs)

An MDP M specifies a dynamical environment an agent interacts with by a 5-tuple $M = \langle S, A, P, R, \gamma \rangle$, where S is the state space, A is the action space, $P : S \times A \times S \to [0, 1]$ is the transition probability function, $R : S \times A \to \mathbb{R}$ is the expected reward function, and γ is the discount factor. Actual rewards obtained by the agent can be stochastic and are assumed to have bounded support $[0, R_{\max}]$. The agent's goal is to optimize its expected value, which is the sum of discounted rewards.

A mapping $\pi : S \to A$ is called a (deterministic and stationary) policy, and specifies the way the agent behaves. We use $V^{\pi}_{M}(s)$ to denote the expected value of behaving according to policy π starting in state s. The policy that maximizes the value function V^{π}_{M} for all states is called an optimal policy of M, denoted π^{*}_{M}, and its value function is abbreviated as V^{*}_{M}. Given the model M, the optimal policy can be found via dynamic programming using the Bellman optimality equation, namely

$$V^{*}_{M}(s) = \max_{a \in A} Q^{*}_{M}(s, a), \tag{1}$$

$$Q^{*}_{M}(s, a) = R(s, a) + \gamma \left\langle P(s, a, \cdot), V^{*}_{M}(\cdot) \right\rangle, \tag{2}$$

where $\langle \cdot, \cdot \rangle$ denotes vector dot product and Q^{π}_{M} is called a Q-value function.

2.2. Abstractions for Model-Based RL

A state abstraction h is a mapping from the primitive state space S to an abstract state space $h(S)$. We use $h(s) \in h(S)$ to denote the abstract state that contains a particular primitive state s. Following certainty equivalence, we assume that the agent builds a model M^{h}_{D} from a dataset D under abstraction h, and then follows the optimal policy for M^{h}_{D}.

Data The dataset D is a set of four-tuples (s, a, r, s'), collected by repeatedly and independently sampling a state-action pair (s, a) from some fixed distribution p fully supported over $S \times A$ (i.e., $p(s, a) > 0 \; \forall s, a$), and then, given (s, a), sampling a reward r from R and a next state s' from P. If some fixed exploration policy is used to collect data, then p will correspond to the state-action occupancy distribution (though the samples will not be strictly independent in this case). For $x \in h(S)$, we denote by $D_{x,a}$ the restriction of D to tuples whose first two elements are $s \in h^{-1}(x)$ and a; that is, $D_{x,a}$ is the portion of the dataset concerning abstract state x and action a.

Model The model estimated from dataset D using abstraction h is $M^{h}_{D} = \langle h(S), A, P^{h}_{D}, R^{h}_{D}, \gamma \rangle$, where $P^{h}_{D}(x, a, x')$ is the empirical likelihood of the transition $(x, a) \to x'$, for $x, x' \in h(S)$ and $a \in A$, and $R^{h}_{D}(x, a)$ is the empirical average reward in (x, a). When referring to the model constructed using the primitive state space, we use the notation M_D, omitting the superscript.

2.3. Problem Statement

Our goal is to choose an abstraction h from a candidate set $\mathcal{H}$ so as to minimize the loss of the optimal policy for M^{h}_{D}:

$$\text{Loss}(h, D) = \left\| V^{*}_{M} - V^{\pi^{*}_{M^{h}_{D}}}_{M} \right\|_{\infty}. \tag{3}$$

Note that $\pi^{*}_{M^{h}_{D}}$ is a mapping from $h(S)$ to A, and has to be *lifted* as $[\pi^{*}_{M^{h}_{D}}]_{M} : s \mapsto \pi^{*}_{M^{h}_{D}}(h(s))$ to be evaluated in M. For notational simplicity, we will not distinguish an abstract policy from its lifted version as long as there is no confusion.

For most of the paper we will be concerned with the following assumption. Later we will discuss how to extend our algorithm and analysis to a more general setting.

Assumption 1. $\mathcal{H} = \{h_c, h_f\}$, where finer abstraction h_f is a refinement of coarser abstraction h_c, i.e., $h_f(s) = h_f(s') \Rightarrow h_c(s) = h_c(s'), \forall\, s, s' \in S$.

3. Bounding the Loss of a Single Abstraction

Before proceeding to describe our solution to the abstraction selection problem, we first establish an upper bound on $\mathrm{Loss}(h, D)$ for any fixed abstraction h. This will allow us to compare the results of our selection algorithm to the loss bounds we could achieve if the qualities of the abstractions were known in advance. Abstraction quality is characterized by the following definitions.

Definition 1. Let $M^h = \langle h(S), A, P^h, R^h, \gamma \rangle$, where, for all $x, x' \in h(S)$ and $a \in A$,

$$P^h(x, a, x') = \frac{\sum_{s \in h^{-1}(x)} p(s, a) \sum_{s' \in h^{-1}(x')} P(s, a, s')}{\sum_{s \in h^{-1}(x)} p(s, a)},$$

$$R^h(x, a) = \frac{\sum_{s \in h^{-1}(x)} p(s, a) R(s, a)}{\sum_{s \in h^{-1}(x)} p(s, a)}.$$

Then M^h is said to be an *approximate homomorphism* of M with *transition error* and *reward error*

$$\epsilon_T^h = \max_{s \in S, a \in A} \sum_{x' \in h(S)} \left| P^h(h(s), a, x') - \sum_{s' \in h^{-1}(x')} P(s, a, s') \right|,$$

$$\epsilon_R^h = \max_{s \in S, a \in A} \left| R^h(h(s), a) - R(s, a) \right|.$$

If $\epsilon_T^h = \epsilon_R^h = 0$, M^h is said to be a (perfect) homomorphism of M, and it is known that $\pi^*_{M^h}$ is an optimal policy for M. As ϵ_T^h and ϵ_R^h increase, $\pi^*_{M^h}$ may incur more loss.

Theorem 1 improves upon and tightens existing bounds from the literature on approximate homomorphisms[1] and bisimulation (e.g., Ravindran & Barto, 2004). (Paduraru et al. (2008) proved a bound tighter than ours by a factor of $1/(1-\gamma)$, but required an asymptotic assumption that $n^h(D)$ is sufficiently large.)

Theorem 1 (Loss bound for a single abstraction)**.** *For any abstraction h, $\forall \delta \in (0, 1)$, w.p. $\geq 1 - \delta$,*

$$\mathrm{Loss}(h, D) \leq \frac{2}{(1-\gamma)^2} \big(\mathrm{Appr}(h) + \mathrm{Estm}(h, D, \delta) \big)$$

where

$$\mathrm{Appr}(h) = \epsilon_R^h + \frac{\gamma R_{\max} \epsilon_T^h}{2(1-\gamma)},$$

$$\mathrm{Estm}(h, D, \delta) = \frac{R_{\max}}{1-\gamma} \sqrt{\frac{1}{2 n^h(D)} \log \frac{2|h(S)||A|}{\delta}},$$

$$n^h(D) = \min_{x \in h(S), a \in A} |D_{x,a}|.$$

The proof is deferred to Appendix A[2]. The bound consists of two terms, where the first increases with the approximation parameters $(\epsilon_T^h, \epsilon_R^h)$ but is independent of the dataset D, and the second has no dependence on ϵ_T^h or ϵ_R^h, but depends on the abstraction via $n^h(D)$, the minimal number of visits to any abstract state-action pair, and $|h(S)|$. The first term is small for accurate abstractions (which have small $(\epsilon_T^h, \epsilon_R^h)$), while the second term is small for compact abstractions (which have small $|h(S)|$ and large $n^h(D)$).

Our goal in this paper is to select from the candidate set $\mathcal{H}$ an abstraction achieving the lowest loss, and we can use the bound in Theorem 1 as a proxy for that loss. (This is a common approach in existing work on abstraction selection as well as machine learning in general; see Section 5 for details.) If the size of the dataset is very small, the bound suggests that we should select a coarse abstraction to reduce estimation error. However, as the size of D grows, $n^h(D)$ increases, and the second term goes to zero while the first remains constant, implying that finer and finer abstractions will in general become preferable (see Jiang et al. (2014) for an empirical illustration). Under Assumption 1, then, the crucial question is: How much data should we require before selecting h_f over h_c?

If ϵ_T^h and ϵ_R^h were known for both abstractions, we could simply calculate an appropriate boundary from Theorem 1. However, in practice, ϵ_T^h and ϵ_R^h are unknown. Nevertheless, we will show that our algorithm can approximately estimate this boundary from data. In particular, we will use D to statistically test whether $Q^*_{M^{h_f}}$ and $Q^*_{M^{h_c}}$ are equal (when lifted); in general, we will reject this hypothesis after we obtain a sufficient amount of data. Perhaps surprisingly, our analysis shows that the point at which this rejection first occurs is almost the same (in the appropriate technical sense) as the point at which h_f becomes preferable to h_c (see Figure 1 for an illustration). Thus, we will use this hypothesis test to define a simple algorithm for abstraction selection that is near-optimal with respect to Theorem 1.

4. Proposed Algorithm and Theoretical Analysis

Before proposing our algorithm, we first define the operators B_D^h and B^h.

[1] In general, approximate homomorphisms can incorporate action aggregation/permutation, but in this paper we only consider aggregation in the state space.

[2] All appendices mentioned in this paper are included in an extended version available at https://sites.google.com/a/umich.edu/nanjiang/icml2015-abstraction.pdf.

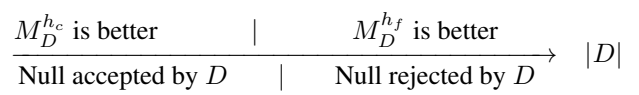

Figure 1. **Upper part:** The preferred abstraction changes as dataset size grows beyond some threshold. **Lower part:** Our algorithm uses the dataset to perform a hypothesis test; when dataset size exceeds some threshold, the null hypothesis will be rejected. We show that the two thresholds have bounded difference, regardless of h_c and h_f.

Definition 2. Given dataset D and abstraction h, $B_D^h : \mathbb{R}^{S\times A} \to \mathbb{R}^{S\times A}$ is defined as follows. For any Q-value function $Q \in \mathbb{R}^{S\times A}$,

$$(B_D^h Q)(s,a) = \frac{\sum_{(r,s')\in D_{h(s),a}} (r + \gamma V_Q(s'))}{|D_{h(s),a}|},$$

where $V_Q(s') = \max_{a'\in A} Q(s',a')$. We define B^h as

$$(B^h Q)(s,a) = \frac{\sum_{s':h(s')=h(s)} p(s',a)\cdot (BQ)(s',a)}{\sum_{s':h(s')=h(s)} p(s',a)},$$

where B is the Bellman optimality operator for M, namely $(BQ)(s,a) = R(s,a) + \gamma \langle P(s,a,\cdot), V_Q(\cdot)\rangle$.

The operator B_D^h is a variation of the Bellman optimality operator for M_D^h, and B^h is the same for M^h. It is not hard to verify that $[Q^*_{M_D^h}]_M$ and $[Q^*_{M^h}]_M$ are, respectively, fixed points of B_D^h and B^h (recall that $[\cdot]_M$ is the lifting operation).

With these definitions, we propose Algorithm 1. It computes a particular statistic using D, and then selects h_f if and only if the statistic exceeds a threshold.

Algorithm 1 ComparePair($D, \mathcal{H}, \delta$)

assert $\mathcal{H} = \{h_c, h_f\}$ satisfies Assumption 1
let $Q = [Q^*_{M_D^{h_c}}]_M$
if

$$\left\| B_D^{h_f} Q - Q \right\|_\infty \geq 2\,\mathrm{Estm}(h_f, D, \delta/3) \qquad (4)$$

then output h_f, **else output** h_c

4.1. Intuition of the Algorithm

Before formally analyzing Algorithm 1, we first present an intuitive explanation for its behavior and show that it makes sensible decisions in various scenarios. The central idea is to statistically test whether

$$[Q^*_{M^{h_f}}]_M = [Q^*_{M^{h_c}}]_M, \qquad (5)$$

which is equivalent (see Lemma 1 and Appendix C) to

$$\left\| B^{h_f} [Q^*_{M^{h_c}}]_M - [Q^*_{M^{h_c}}]_M \right\|_\infty = 0. \qquad (6)$$

The LHS of Equation (6) is effectively the Bellman residual of $Q^*_{M^{h_c}}$ when treating M^{h_f} as the true model. Since the required quantities are not known in advance, we approximate them from data and check whether the measured error exceeds a positive rejection threshold. This gives the selection criterion of Equation (4).

Consider two extreme cases. First, when M^{h_c} is a perfect homomorphism of M^{h_f}, Equation (5) always holds and we never reject the null hypothesis, thus our algorithm always returns h_c. This makes sense, since the abstractions have equal approximation error but h_c has lower estimation error. On the other hand, when Equation (5) does not hold, given enough data our test will reject the null hypothesis and select h_f. Again, this is sensible since h_f has lower approximation error, and in the limit of data the estimation error for both abstractions is zero.

Of course, the usual situation is that Equation (5) does not hold but D is finite. Suppose in this case that M^{h_f} is a perfect homomorphism of M; then Algorithm 1 can be seen as approximately comparing the bound in Theorem 1 for h_f and h_c, as follows. Since $\mathrm{Appr}(h_f) = 0$ and the estimation errors are computable from known quantities, the only unknown quantity needed for this comparison is $\mathrm{Appr}(h_c)$. In principle, $\mathrm{Appr}(h_c)$ is a function of M and M^{h_c}, and could be approximated from data using M_D and $M_D^{h_c}$; however, the estimate of M_D will be poor when $|S|$ is large (which is why we require abstraction in the first place). Instead, since h_f is exact by assumption, we can compare M^{h_c} directly to M^{h_f}. The LHS of Equation (4) provides this estimate of $\mathrm{Appr}(h_c)$; see the left panel of Figure 2 for a visual illustration.

In the most general scenario, where the dataset is finite and *both* abstractions are approximate, we need a reliable estimate of $\mathrm{Appr}(h_c) - \mathrm{Appr}(h_f)$ to make the comparison using Theorem 1, but we no longer have a statistically efficient way of estimating $\mathrm{Appr}(h_f)$ or $\mathrm{Appr}(h_c)$. However, our analysis shows that even when M^{h_f} is not homomorphic to M, the three models can be seen as roughly "on the same line", as visualized in the right panel of Figure 2. As a result, we can use the dashed line—a measure of distance between M^{h_f} and M^{h_c}—to approximate the desired difference between the solid lines. This idea is the basis for Lemma 4, which is a key ingredient in the theoretical guarantee for Algorithm 1.

4.2. Theoretical Analysis

We next state the formal guarantee of our algorithm.

Theorem 2. *Given dataset D, if $\mathcal{H}$ satisfies Assumption 1, the loss of the abstraction selected by Algorithm 1 is*

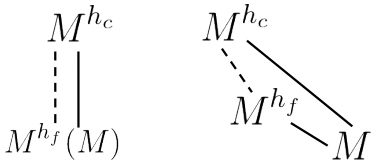

Figure 2. **Left panel:** When M^{h_f} is a perfect homomorphism of M, we can obtain the true approximation error of M^{h_c} (solid line) by computing its approximation error w.r.t. M^{h_f} (dashed line). The two notions of approximation are equivalent, but the latter is statistically easier to estimate. **Right panel:** When M^{h_f} is also approximate, our theoretical analysis shows that M, M^{h_f}, and M^{h_c} are always roughly "on the same line", so that the approximation error of M^{h_c} w.r.t. M^{h_f} (dashed line) is a good proxy for the difference between the true approximation errors of M^{h_c} and M^{h_f} (solid lines).

bounded by

$$\frac{2}{(1-\gamma)^2}\min\Big\{\operatorname{Appr}(h_f)+\frac{3-\gamma}{1-\gamma}\operatorname{Estm}(h_f,D,\delta/3),\ \frac{3-\gamma}{1-\gamma}\operatorname{Appr}(h_c)+\frac{1+\gamma}{1-\gamma}\operatorname{Estm}(h_c,D,\delta/3)\Big\} \quad (7)$$

with probability at least $1-\delta$.

Equation (7) is the minimum of two terms. The first is nearly (up to a factor of $O(1/(1-\gamma))$) the loss bound of h_f using Theorem 1, and the second is nearly the loss bound of h_c. Recall that Theorem 1 is our proxy for loss; therefore, the loss bound for Algorithm 1 is as good as the loss bound of the *better* abstraction up to a factor linear in $1/(1-\gamma)$. Compared to Theorem 1, the estimation error terms in Equation (7) have increased from $\operatorname{Estm}(\cdot,\cdot,\delta)$ to $\operatorname{Estm}(\cdot,\cdot,\delta/3)$; however, this has little influence as $\operatorname{Estm}(\cdot,\cdot,\delta)$ depends only square-root logarithmically on $1/\delta$.

Claim 1 (Theorem 2 is near-optimal w.r.t. Theorem 1)**.** *Equation (7) is at most the minimum of the bound in Theorem 1 as applied to h_f and to h_c, up to a factor of $O(\frac{1}{1-\gamma})$.*

We will prove Theorem 2 with the help of the following lemmas. Their proofs are deferred to Appendices A and B.

Lemma 1. *For any Bellman optimality operators B_1, B_2 (both operating on $\mathbb{R}^{S\times A}$ and having contraction rate γ), letting Q_1 and Q_2 be their respective fixed points, we have*

$$\|Q_1-Q_2\|_\infty \le \frac{\|B_1Q_2-Q_2\|_\infty}{1-\gamma}.$$

Lemma 2. *Consider B^h_D as defined in Definition 2. For any $h\in\mathcal{H}$ and deterministic $Q:\mathbb{R}^{S\times A}$ with bounded range $[0,R_{\max}/(1-\gamma)]$, w.p. $\ge 1-\delta$,*

$$\left\|B^h_DQ-B^hQ\right\|_\infty \le \operatorname{Estm}(h,D,\delta).$$

Lemma 3. *Let B be the Bellman optimality operator of M. For any $Q:\mathbb{R}^{h(S)\times A}$ with bounded range $[0,R_{\max}/(1-\gamma)]$, we have*

$$\left\|B[Q]_M-B^h[Q]_M\right\|_\infty \le \operatorname{Appr}(h).$$

Lemma 4. *$\forall Q:\mathbb{R}^{S\times A}$ with bounded range $[0,R_{\max}/(1-\gamma)]$,*

$$\begin{aligned}&\|BQ-B^{h_c}Q\|_\infty\\ &\quad\le \left\|BQ-B^{h_f}Q\right\|_\infty+\left\|B^{h_f}Q-B^{h_c}Q\right\|_\infty\\ &\quad\le 3\left\|BQ-B^{h_c}Q\right\|_\infty.\end{aligned}$$

We briefly sketch the proof of Theorem 2 before proceeding to the details. Recall that our goal is to determine which abstraction has a smaller loss bound according to Theorem 1; that is, we want to check whether

$$\operatorname{Appr}(h_c)-\operatorname{Appr}(h_f) \ge \operatorname{Estm}(h_f,D,\delta)-\operatorname{Estm}(h_c,D,\delta),$$

where the LHS is unknown. To approximate it, we first use Lemma 4, which implies that

$$\begin{aligned}&\|B[Q^*_{M^{h_c}}]_M-[Q^*_{M^{h_c}}]_M\|_\infty &(8)\\ \approx\ &\left\|B[Q^*_{M^{h_c}}]_M-B^{h_f}[Q^*_{M^{h_c}}]_M\right\|_\infty &(9)\\ &+\left\|B^{h_f}[Q^*_{M^{h_c}}]_M-[Q^*_{M^{h_c}}]_M\right\|_\infty. &(10)\end{aligned}$$

Expression (10) is a quantity closely related to the statistic computed by our algorithm (see Equation (6)), so to establish that the statistic is a good proxy for $\operatorname{Appr}(h_c)-\operatorname{Appr}(h_f)$, we will show that

$$\operatorname{Appr}(h_c)-\operatorname{Appr}(h_f) \approx \text{Expression (8)}-\text{Expression (9)}.$$

Expression (8) is easy to deal with, as the Bellman residual of $[Q^*_{M^{h_c}}]_M$ is a better characterization of the approximation error of h_c than $\operatorname{Appr}(h_c)$.[3] Expression (9) is a bit trickier: we know it is not an overestimate, as Lemma 3 guarantees that it is upper bounded by $\operatorname{Appr}(h_f)$. However, there exists the risk of underestimation: for instance, if h_c aggregates all primitive states into a single abstract state, then $[Q^*_{M^{h_c}}]_M$ is a constant function and Expression (9) only reflects the reward error of h_f, and will not change regardless of the transition error.

[3] In this discussion we do not strictly distinguish between approximate homomorphism ($\operatorname{Appr}(h)$) and approximate Q^*-irrelevance (the Bellman residual of $Q^*_{M^h}$) in characterizing the approximation error of h. Technical details can be found in proofs and we point the readers to Li et al. (2006) for further reading.

To deal with this, we consider two cases separately. First, when h_c is the better abstraction, we have $[Q^*_{M^{h_c}}]_M \approx Q^*_M$, hence

$$\text{Expression (9)} \approx \left\| BQ^*_M - B^{h_f} Q^*_M \right\|_\infty . \qquad (11)$$

According to Lemma 1, the RHS of Equation (11) is an alternative characterization of the approximation error of h_f, so in this case we will not underestimate too much. On the other hand, when h_f is better, underestimation of its approximation error only biases our selection towards the better abstraction, and is not a concern.

Below we include part of the proof of Theorem 2.

Proof of Theorem 2. Using Lemma 2, w.p. at least $1-\delta$ we have

$$\left\| B_D^{h_f}[Q^*_{M^{h_f}}]_M - B^{h_f}[Q^*_{M^{h_f}}]_M \right\|_\infty \le \text{Estm}(h_f, D, \delta/3),$$

and similar concentration bounds hold for $B_D^{h_c}[Q^*_{M^{h_c}}]_M$ and $B_D^{h_f}[Q^*_{M^{h_c}}]_M$ simultaneously.

Regardless of which abstraction the algorithm selects, we can always bound its loss using Theorem 1, so it suffices to show that we can bound the loss of the selected abstraction in terms of the other. We consider each possibility in turn.

If the algorithm outputs h_c, we can bound the loss of h_c by parameters of h_f:

$$\begin{aligned}
&\text{Loss}(h_c, D) \\
&\le \frac{2}{(1-\gamma)^2} \left\| B[Q^*_{M_D^{h_c}}]_M - [Q^*_{M_D^{h_c}}]_M \right\|_\infty && (12)\\
&\le \frac{2}{(1-\gamma)^2} \Bigg(\left\| B[Q^*_{M_D^{h_c}}]_M - B_D^{h_f}[Q^*_{M_D^{h_c}}]_M \right\|_\infty \\
&\quad + \left\| B_D^{h_f}[Q^*_{M_D^{h_c}}]_M - [Q^*_{M_D^{h_c}}]_M \right\|_\infty \Bigg) && (13)\\
&\le \frac{2}{(1-\gamma)^2} \Bigg(\left\| B[Q^*_{M^{h_c}}]_M - B_D^{h_f}[Q^*_{M^{h_c}}]_M \right\|_\infty \\
&\quad + 2\,\text{Estm}(h_f, D, \delta/3) \\
&\quad + 2\gamma \left\| [Q^*_{M^{h_c}}]_M - [Q^*_{M_D^{h_c}}]_M \right\|_\infty \Bigg) && (14)\\
&\le \frac{2}{(1-\gamma)^2} \Bigg(\left\| B[Q^*_{M^{h_c}}]_M - B^{h_f}[Q^*_{M^{h_c}}]_M \right\|_\infty \\
&\quad + 3\,\text{Estm}(h_f, D, \delta/3) \\
&\quad + \frac{2\gamma}{1-\gamma}\,\text{Estm}(h_c, D, \delta/3) \Bigg) && (15)\\
&\le \frac{2}{(1-\gamma)^2} \left(\text{Appr}(h_f) + \frac{3-\gamma}{1-\gamma}\,\text{Estm}(h_f, D, \delta/3) \right).
\end{aligned}$$

Equation (12) is a standard loss bound using the Bellman residual. In Equation (13), we use the triangle inequality to introduce the statistic computed by our algorithm. In the first term of Equation (14), we replace $[Q^*_{M_D^{h_c}}]_M$ by $[Q^*_{M^{h_c}}]_M$ using the fact that the Bellman operators have contraction rate γ ($\|BQ - BQ'\|_\infty \le \gamma \|Q - Q'\|_\infty$), and in the second term we use the fact that the algorithm chose h_c, and thus Equation (4) did not hold. Next, we apply the probabilistic guarantees stated at the beginning of the proof to remove the D subscripts on operators and Q-value functions, and finally the $\text{Appr}(h_f)$ term appears thanks to Lemma 3.

The remainder of the proof uses similar techniques and appears in Appendix B. □

4.3. Extension to Arbitrary-Size Candidate Sets

We briefly discuss how to extend the above algorithm and analysis to the following setting.

Assumption 2. $\mathcal{H} = \{h_1, \ldots, h_{|\mathcal{H}|}\}$, where h_i is a refinement of h_{i-1} for $i = 2, \ldots, |\mathcal{H}|$.

This is the setting considered by Hallak et al. (2013), and $\mathcal{H} = \{h_c, h_f\}$ is the special case where $|\mathcal{H}| = 2$. A natural idea is to use Algorithm 1 as a subroutine, successively comparing the best abstraction seen so far with the remaining elements in $\mathcal{H}$ in some order. The crucial questions are: (1) in what order should we examine the abstractions (e.g., coarse-to-fine, fine-to-coarse, or a random/adaptive order), and (2) can we adapt the analysis in Section 4.2 to show that the selected abstraction is still near-optimal w.r.t. Theorem 1 for larger $\mathcal{H}$? It turns out that, if we examine abstractions in order from coarse to fine, near-optimality is preserved. Algorithm 2 provides a detailed specification for the process, and Theorem 3 gives the resulting guarantee.

Algorithm 2 CompareSequence($D, \mathcal{H}, \delta$)

```
assert H = {h_1, ..., h_|H|} satisfies Assumption 2
let ĥ = h_1    // start with the coarsest abstraction
for i=2 to |H| do
  ĥ = ComparePair(D, {ĥ, h_i}, 2δ/|H|^2)
end for
output ĥ
```

Theorem 3. *If $\mathcal{H}$ satisfies Assumption 2 and has constant size, then Algorithm 2 is near-optimal w.r.t. Theorem 1, i.e., the loss of the selected abstraction is bounded w.p. at least $1-\delta$ by*

$$\min_{h \in \mathcal{H}} \left(\text{Appr}(h) + \text{Estm}\left(h, D, 2\delta/(3|\mathcal{H}|^2)\right) \right)$$

up to a factor polynomial in $1/(1-\gamma)$.

The biggest challenge in generalizing our analysis to the case of $|\mathcal{H}| > 2$ is that the two sides of Equation (7) have different semantics—that is, the LHS is loss, while the RHS is approximation/estimation error. This means that successive comparisons cannot (naively) apply the bound transitively. Recall that, in the proof of Theorem 2, we considered the selection of h_c and the selection of h_f separately. It turns out that we can modify the analysis to obtain consistent, transitive semantics, but only for the case where h_c is selected. This is enough for near-optimality as long as we order the abstractions from coarse to fine, avoiding the bad case of problematic abstractions. For a more detailed discussion and a proof sketch of Theorem 3, see Appendix C.

5. Related Work

In this section we review prior theoretical work that is relevant to the abstraction selection problem. The discussion is summarized in Table 1. For recent empirical advances on the problem, we refer the reader to Konidaris & Barto (2009), Cobo et al. (2014), and Seijen et al. (2014).

5.1. Hypothesis Test Based Algorithms

Jong & Stone (2005) (row 1) considered the factored MDP setting, where state is determined by a vector of features and an abstraction is a subset of those features. They proposed a selection procedure that statistically tests whether the optimal policy depends on certain features, aiming to aggregate states having the same optimal action and thus create a π^*-*irrelevance* abstraction. However, π^*-irrelevance abstractions can yield sub-optimal policies when applying Q-learning even with infinite data, so this method is not statistically consistent (Li et al., 2006, Theorem 4).

Hallak et al. (2013) (row 2), in the work most closely related to ours, considered the setting of Assumption 2 and suggested comparing h_c and h_f by statistically testing whether M^{h_c} is a perfect homomorphism of M^{h_f} using D. They showed theoretically that their procedure will *asymptotically* identify any abstraction that is a *perfect* homomorphism of M. However, if all the candidate abstractions are approximate, or the dataset is finite, their analysis does not apply.

Nevertheless, there are interesting similarities between our Algorithm 1 and the method of Hallak et al. (2013): both algorithms test *relative* properties of h_c and h_f so as to avoid the large primitive representation, and both choose the coarser abstraction unless a statistical test rejects the null hypothesis that h_c and h_f are (in some sense) equivalent. However, our analysis shows that this type of algorithm can still have provable guarantees even when the data are insufficient and the abstractions are approximate—in fact, it can be near-optimal with respect to a loss bound.

There are several important technical differences between our algorithm and that of Hallak et al. (2013): (1) We use Q^*-*irrelevance* as the equivalence criterion in our hypothesis test, whereas they use homomorphism; Q^*-irrelevance is a strictly more general relationship than homomorphism (Li et al., 2006, Theorem 2) that avoids the problematic L_1 norm as a characterization of estimation error (Maillard et al., 2014) and enables convenient mathematical tools for finite sample analysis (e.g., the B_D^h operator). (2) We fully specify the rejection threshold for the hypothesis test (up to the probability guarantee δ) without introducing additional hyperparameters, while in their work the rate of threshold decay as the dataset grows is left to the practitioner. This choice can have a significant impact on the transient behavior of the algorithm.

5.2. Reduction to Offline Policy Evaluation

Inspired by model selection techniques in supervised learning, abstractions can also be selected using a cross-validation procedure: if a second dataset D' is given independently of D, then we can evaluate the policies computed under different abstractions from D (i.e., $\{\pi^*_{M_D^h} : h \in \mathcal{H}\}$) on D'. This turns the abstraction selection problem into an offline policy evaluation problem, and the loss guarantee depends entirely on the accuracy of the offline evaluation estimator. Below we briefly discuss two commonly used estimators; see more details in Paduraru (2013).

5.2.1. Importance Sampling Estimator

When D' comprises trajectories sampled according to a known stochastic policy, importance sampling (row 3) can be used to obtain an unbiased estimate of the value, with respect to the initial state distribution in the data, of the policy under evaluation (Precup et al., 2000). The variance of this estimate has no dependence on the size of the primitive state space, but in general has exponential dependence on the horizon (Mandel et al., 2014): to evaluate a deterministic policy π, the estimator must restrict itself to trajectories in which the action choice at each time step exactly agrees with π. If, for instance, the sampled trajectories are $L = 1/(1-\gamma)$ steps long and generated by choosing uniformly random actions, then the probability of any single trajectory being useful is $1/|A|^L$, hence the proportion of data available for estimation decreases exponentially with L. Even so, importance sampling can be practical and has been successfully applied to real-world problems with very short horizons (Li et al., 2011; Mandel et al., 2014).

5.2.2. Model-Based Estimator

For problems with longer horizons, an alternative is the model-based estimator (row 4), which uses D' to construct a model $M_{D'}$ (without abstraction) and then evaluates a policy π by computing $V^\pi_{M_{D'}}$. However, this approach is not

Table 1. Comparison of algorithms that can be applied to the abstraction selection problem. If an entry exhibits a desired property (which we judge by generality and practicality), we mark it as **bold**. In the first row we provide the properties of model-based RL with primitive representation as a baseline to compare against.

	Finite Sample Guarantee		Assumption on Candidate Abstractions	Tuning Hyperparameters	Optimization Objective
	Dependence on Representation	Dependence on Horizon			
No Abstraction	$\|S\|$	**Polynomial**	-	-	-
1.Jong & Stone (2005)	[No guarantee]		**Subsets of state features**	**No**[a]	-
2.Hallak et al. (2013)	[Only asymptotic guarantee]		Successive refinements	Statistical test threshold as a function of sample size	Coarseness of perfect homomorphisms
3.Importance Sampling	**No**	Exponential	**No**	**No**	**Loss**
4.Model-based Estimator	$\|S\|$	**Polynomial**	**No**	**No**	**Loss**
5.Farahmand & Szepesvári (2011)	Size of regressor abstraction	**Polynomial**	**No**	Choice of regressor abstraction	Bellman residual loss bound
Our method	**Size of best abstraction**	**Polynomial**	Successive refinements	**No**	Approximate homomorphism loss bound

[a]Their algorithm has a single parameter which is the p-value threshold for hypothesis test, and they suggest using 0.05 in practice. In fact, all the methods listed in this table except the first 3 rows require a similar confidence level parameter.

useful in our setting, since $M_{D'}$ is itself hard to estimate: the bound on the estimation error $\left\|V^{\pi}_{M} - V^{\pi}_{M_{D'}}\right\|_{\infty}$ depends on the minimal state visitation number, and therefore on $|S|$ (Mannor et al., 2007; Paduraru et al., 2008).

Alternatively, the validation model can be estimated under an abstraction to avoid the dependence on $|S|$, but this solution is circular: if we knew a good abstraction for policy evaluation, we could have used it to obtain a good policy in the first place. For instance, Farahmand & Szepesvári (2011) (row 5) proposed an offline policy evaluation procedure that selects value functions (from which policies are computed) based on their estimated Bellman residuals, which are estimated with the help of an additional regressor that learns BQ from data for the candidate Qs. The theoretical guarantee for this method depends on the accuracy of the regressor (see their Theorem 2, especially the dependence on $\bar{b}_k$). For the reason noted above, this is problematic in our setting: the abstractions are themselves regressors (where $B^h_D Q$ is the function being learned), so if we knew how to select a good abstraction for regression, then the same one could have been used to learn a policy instead.

5.3. The Online Setting

Ortner et al. (2014) proposed a representation (abstraction) selection algorithm in the online exploration and exploitation setting that tests whether a representation faithfully predicts the return of a roll-out trajectory. Their regret bound depends on the *sum* of sizes of the state spaces for all representations under consideration (see their Theorem 3). While the online setting has additional complications, in our offline setting this bound is loose and can be improved simply by selecting the finest available abstraction. On the other hand, although our algorithm assumes structure in the candidate abstractions (they must be successive refinements), our loss bound depends only on the *best* abstraction.

6. Conclusion

In this paper we considered the abstraction selection problem in the setting where the amount of data is limited and candidate abstractions may all be approximate. As far as we know, there is no provable algorithm that achieves significantly better dependence on representation (than the baseline of the primitive state space) without blowing up the dependence on horizon exponentially, suggesting that the problem is generally hard. Our work showed that, when the candidate abstractions satisfy certain refinement assumptions, a simple hypothesis test based algorithm can select abstractions with a loss bound only depending on the best abstraction up to a factor polynomial in horizon. The theoretical analysis is fundamentally based on the approximation/estimation error trade-off with finite data, departing from the asymptotic analysis of previous work that only considered perfect abstractions in the limit of data. Possible directions for future work include relaxing the assumption (e.g., to the setting of feature selection), and developing heuristic algorithms based on the algorithmic ideas provided in this paper.

Acknowledgement

This work was supported by NSF grant IIS-1319365. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the views of the sponsors.

References

Cobo, Luis C, Subramanian, Kaushik, Isbell, Charles L, Lanterman, Aaron D, and Thomaz, Andrea L. Abstraction from demonstration for efficient reinforcement learning in high-dimensional domains. *Artificial Intelligence*, 216: 103–128, 2014.

Dinculescu, Monica and Precup, Doina. Approximate predictive representations of partially observable systems. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 895–902, 2010.

Even-Dar, Eyal and Mansour, Yishay. Approximate equivalence of Markov decision processes. In *Learning Theory and Kernel Machines*, pp. 581–594. 2003.

Farahmand, Amir-massoud and Szepesvári, Csaba. Model selection in reinforcement learning. *Machine learning*, 85(3):299–332, 2011.

Hallak, Assaf, Di-Castro, Dotan, and Mannor, Shie. Model selection in markovian processes. In *Proceedings of the 19th ACM SIGKDD Conference on Knowledge Discovery and Data mining*, pp. 374–382, 2013.

Jiang, Nan, Singh, Satinder, and Lewis, Richard. Improving UCT planning via approximate homomorphisms. In *Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems*, pp. 1289–1296, 2014.

Jong, Nicholas K and Stone, Peter. State abstraction discovery from irrelevant state variables. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pp. 752–757, 2005.

Konidaris, George and Barto, Andrew. Efficient Skill Learning using Abstraction Selection. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, 2009.

Lever, Guy, Baldassarre, Luca, Gretton, Arthur, Pontil, Massimiliano, and Grünewälder, Steffen. Modelling transition dynamics in MDPs with RKHS embeddings. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp. 535–542, 2012.

Li, Lihong, Walsh, Thomas J, and Littman, Michael L. Towards a unified theory of state abstraction for MDPs. In *Proceedings of the 9th International Symposium on Artificial Intelligence and Mathematics*, pp. 531–539, 2006.

Li, Lihong, Chu, Wei, Langford, John, and Wang, Xuanhui. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, pp. 297–306, 2011.

Maillard, Odalric-Ambrym, Mann, Timothy A, and Mannor, Shie. "How hard is my MDP?" The distribution-norm to the rescue. In *Advances in Neural Information Processing Systems*, pp. 1835–1843, 2014.

Mandel, Travis, Liu, Yun-En, Levine, Sergey, Brunskill, Emma, and Popovic, Zoran. Offline policy evaluation across representations with applications to educational games. In *Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems*, pp. 1077–1084, 2014.

Mannor, Shie, Simester, Duncan, Sun, Peng, and Tsitsiklis, John N. Bias and variance approximation in value function estimates. *Management Science*, 53(2):308–322, 2007.

Ormoneit, Dirk and Sen, Śaunak. Kernel-based reinforcement learning. *Machine learning*, 49(2-3):161–178, 2002.

Ortner, Ronald, Maillard, Odalric-Ambrym, and Ryabko, Daniil. Selecting Near-Optimal Approximate State Representations in Reinforcement Learning. *arXiv preprint arXiv:1405.2652*, 2014.

Paduraru, Cosmin. *Off-policy Evaluation in Markov Decision Processes*. PhD thesis, McGill University, 2013.

Paduraru, Cosmin, Kaplow, Robert, Precup, Doina, and Pineau, Joelle. Model-based reinforcement learning with state aggregation. In *8th European Workshop on Reinforcement Learning*, 2008.

Parr, Ronald, Li, Lihong, Taylor, Gavin, Painter-Wakefield, Christopher, and Littman, Michael L. An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 752–759, 2008.

Precup, Doina, Sutton, Richard S, and Singh, Satinder. Eligibility traces for off-policy policy evaluation. In *Proceedings of the 17th International Conference on Machine Learning*, pp. 759–766, 2000.

Ravindran, Balaraman. *An algebraic approach to abstraction in reinforcement learning*. PhD thesis, University of Massachusetts Amherst, 2004.

Ravindran, Balaraman and Barto, Andrew. Approximate homomorphisms: A framework for nonexact minimization in Markov decision processes. In *Proceedings of the 5th International Conference on Knowledge-Based Computer Systems*, 2004.

Seijen, Harm, Whiteson, Shimon, and Kester, Leon. Efficient abstraction selection in reinforcement learning. *Computational Intelligence*, 30(4):657–699, 2014.

Singh, Satinder and Yee, Richard. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16(3):227–233, 1994.

Sutton, Richard and Barto, Andrew. *Introduction to reinforcement learning*. MIT Press, 1998.

Sutton, Richard S, McAllester, David A, Singh, Satinder P, and Mansour, Yishay. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 99, pp. 1057–1063, 1999.

Talvitie, Erik and Singh, Satinder. Learning to make predictions in partially observable environments without a generative model. *Journal of Artificial Intelligence Research (JAIR)*, 42:353–392, 2011.

A. Proof of Theorem 1

We first prove Lemma 1, 2 and 3.

Proof of Lemma 1. $\forall s \in S, a \in A$,

$$\begin{aligned}&\|Q_1 - Q_2\|_\infty \\ =\ &\|B_1 Q_1 - B_1 Q_2 + B_1 Q_2 - Q_2\|_\infty \\ \le\ &\gamma \|Q_1 - Q_2\|_\infty + \|B_1 Q_2 - Q_2\|_\infty .\end{aligned}$$

Hence the bound follows. Note that this result subsumes the standard Bellman residual bound, when we let Q_2 be an approximate Q-value function (e.g. $[Q^*_{M^h}]_M$, where $B_2 = B^h$), and Q_1 be the true optimal value function Q^*_M (where $B_1 = B$). Furthermore, thanks to the definition of B^h, we can use this bound in an alternative form, namely bounding $\left\|Q^*_M - [Q^*_{M^h}]_M\right\|_\infty$ by $\left\|Q^*_M - B^h Q^*_M\right\|_\infty$. We will use both forms (and sometimes treating M^h as the true model) throughout the theoretical analysis depending on the context. □

Proof of Lemma 2. According to Definition 2, $(B^h_D Q)(s,a)$ is the average of $r + \gamma V_Q(s')$ for $(r, s') \in D_{h(s),a}$, which are independent random variables with bounded range $[0, R_{\max}/(1-\gamma)]$. When $|D_{h(s),a}| > 0$,[4] it is straight-forward to verify that for any deterministic Q, $(B^h Q)(s,a) = \mathbb{E}_D\big\{(B^h_D Q)(s,a) \;\big|\; |D_{h(s),a}| > 0\big\}$. Hence, Hoeffding's inquality applies, $\forall t > 0$,

$$\begin{aligned}&\mathbb{P}_D\Big\{\Big|(B^h_D Q)(s,a) - (B^h Q)(s,a)\Big| \ge t\Big\} \\ \le\ &2\exp\left(-\frac{2t^2|D_{x,a}|}{R^2_{\max}/(1-\gamma)^2}\right).\end{aligned}$$

Now we find t that makes the inequality hold for all $(s,a) \in S \times A$ simultaneously w.p. at least $1-\delta$ via union bound. Note, however, that $B^h_D Q$ (and $B^h Q$) takes constant value among states aggregated by h, hence we only have $|h(S)||A|$ events in the union bound instead of $|S||A|$ ones. The t that satisfies our requirement turns out to be

$$t = \frac{R_{\max}}{1-\gamma}\sqrt{\frac{1}{2n^h(D)}\log\frac{2|h(S)||A|}{\delta}} = \mathrm{Estm}(h, D, \delta),$$

and this completes the proof. □

Proof of Lemma 3. $\forall s \in S, a \in A$,

$$\begin{aligned}&\big|(B[Q]_M)(s,a) - (B^h[Q]_M)(s,a)\big| \\ =\ &\Big|R(s,a) + \gamma\Big\langle P(s,a,\cdot), [V_Q]_M(\cdot)\Big\rangle \\ &- R^h(h(s),a) - \gamma\Big\langle P^h(h(s),a,\cdot), V_Q(\cdot)\Big\rangle\Big| \\ =\ &\Big|R(s,a) + \gamma\Big\langle \sum_{s'\in h^{-1}(\cdot)} P(s,a,s'), V_Q(\cdot) - \frac{R_{\max}}{2(1-\gamma)}\Big\rangle \\ &- R^h(h(s),a) - \gamma\Big\langle P^h(h(s),a,\cdot), V_Q(\cdot) - \frac{R_{\max}}{2(1-\gamma)}\Big\rangle\Big| \\ \le\ &\epsilon^h_R + \epsilon^h_T \frac{R_{\max}}{2(1-\gamma)} = \mathrm{Appr}(h).\end{aligned}$$

This completes the proof. □

Proof of Theorem 1. Let $[Q^*_{M^h_D}]_M$ denote $Q^*_{M^h_D}$ lifted to M, namely $[Q^*_{M^h_D}]_M(s) = Q^*_{M^h_D}(h(s))$. We have,

$$\begin{aligned}&\left\|V^*_M - V^{\pi^*_{M^h_D}}_M\right\|_\infty \\ \le\ &\frac{2}{1-\gamma}\left\|Q^*_M - [Q^*_{M^h_D}]_M\right\|_\infty \text{ (Singh \& Yee, 1994)} \\ \le\ &\frac{2}{1-\gamma}\left(\|Q^*_M - [Q^*_{M^h}]_M\|_\infty + \left\|[Q^*_{M^h}]_M - [Q^*_{M^h_D}]_M\right\|_\infty\right) \\ =\ &\frac{2}{1-\gamma}\left(\|Q^*_M - [Q^*_{M^h}]_M\|_\infty + \left\|Q^*_{M^h} - Q^*_{M^h_D}\right\|_\infty\right).\end{aligned}$$

According to Lemma 3, the first term in the bracket can be bounded as:

$$\begin{aligned}\|Q^*_M - [Q^*_{M^h}]_M\|_\infty &\le \frac{\left\|B[Q^*_{M^h}]_M - B^h[Q^*_{M^h}]_M\right\|_\infty}{1-\gamma} \\ &\le \frac{\mathrm{Appr}(h)}{1-\gamma}.\end{aligned}$$

For the second term, we use Lemma 1 by letting $B_1 = B^{h_f}$ and $B_2 = B^{h_f}_D$, and then apply Lemma 2: w.p. at least $1-\delta$,

$$\begin{aligned}\left\|Q^*_{M^h} - Q^*_{M^h_D}\right\|_\infty &\le \frac{\left\|B^h[Q^*_{M^h}]_M - B^h_D[Q^*_{M^h}]_M\right\|_\infty}{1-\gamma} \\ &\le \frac{\mathrm{Estm}(h, D, \delta)}{1-\gamma}.\end{aligned}$$

Combining the bounds for the two terms and the theorem follows. □

B. Proof of Theorem 2

We first prove the remaining Lemma.

Proof of Lemma 4. The left inequality is trivial from the triangular inequality. To prove the right inequality, we bound $\left\|BQ - B^{h_f}Q\right\|_\infty$ and $\left\|B^{h_f}Q - B^{h_c}Q\right\|_\infty$ by $\left\|BQ - B^{h_c}Q\right\|_\infty$ separately. The key is to notice that, for any $x \in h_f(S)$, $(B^{h_f}Q)(x,a)$ is always a convex average of

$$\{(BQ)(s,a) : s \in h_f^{-1}(x)\}.$$

[4] When $|D_{h(s),a}| = 0$, $n^h(D) = 0$ and the RHS of the bound goes to infinity, which promises nothing and is always correct.

We first show $\left\|BQ - B^{h_f}Q\right\|_\infty \leq 2\left\|BQ - B^{h_c}Q\right\|_\infty$. Notice that there exist $s, s' \in S, a \in A$ s.t. $h_f(s) = h_f(s')$ and

$$|(BQ)(s,a) - (BQ)(s',a)| \geq \left\|BQ - B^{h_f}Q\right\|_\infty.$$

Using the same argument on h_c, it is obvious that

$$\begin{aligned}
&\left\|BQ - B^{h_c}Q\right\|_\infty \\
\geq &\max_{\substack{h_c(s)=h_c(s') \\ a\in A}} |(BQ)(s,a) - (BQ)(s',a)| \,/\, 2 \\
\geq &\max_{\substack{h_f(s)=h_f(s') \\ a\in A}} |(BQ)(s,a) - (BQ)(s',a)| \,/\, 2 \\
\geq &\left\|BQ - B^{h_f}Q\right\|_\infty \,/\, 2,
\end{aligned}$$

hence the bound follows.

Next we show $\left\|B^{h_f}Q - B^{h_c}Q\right\|_\infty \leq \left\|BQ - B^{h_c}Q\right\|_\infty$. Consider the state-action pair that achieves the max norm of $\left\|B^{h_f}Q - B^{h_c}Q\right\|_\infty$, i.e.

$$\left|(B^{h_f}Q)(s,a) - (B^{h_c}Q)(s,a)\right| = \left\|B^{h_f}Q - B^{h_c}Q\right\|_\infty.$$

Since $(B^{h_f}Q)(s,a)$ is a convex average of $\{(BQ)(s',a) : h_f(s') = h_f(s)\}$, there always exists $s' : h_f(s') = h_f(s)$ such that $(BQ)(s',a) \geq (B^{h_f}Q)(s,a)$, and $s'' : h_f(s'') = h_f(s)$ such that $(BQ)(s'',a) \leq (B^{h_f}Q)(s,a)$. Note that $(B^{h_c}Q)(s,a) = (B^{h_c}Q)(s',a) = (B^{h_c}Q)(s'',a)$, hence either

$$\left|(BQ)(s',a) - (B^{h_c}Q)(s',a)\right|$$

or

$$\left|(BQ)(s'',a) - (B^{h_c}Q)(s'',a)\right|$$

will be no less than

$$\left|(B^{h_f}Q)(s,a) - (B^{h_c}Q)(s,a)\right|,$$

which implies that

$$\left\|B^{h_f}Q - B^{h_c}Q\right\|_\infty \leq \left\|BQ - B^{h_c}Q\right\|_\infty.$$

This completes the proof. □

Proof of Theorem 2 (continued). Similarly, if the algorithm outputs h_f,

$$\begin{aligned}
&\mathrm{Loss}(h_f, D) \\
\leq &\frac{2}{1-\gamma}\left(\left\|Q^*_M - [Q^*_{M^{h_f}}]_M\right\|_\infty \right. \\
&\left. + \left\|[Q^*_{M^{h_f}}]_M - [Q^*_{M_D^{h_f}}]_M\right\|_\infty\right) \\
\leq &\frac{2}{(1-\gamma)^2}\left(\left\|B^{h_f}Q^*_M - BQ^*_M\right\|_\infty \right. \\
&\left. + \mathrm{Estm}(h_f, D, \delta/3)\right) \qquad (16) \\
\leq &\frac{2}{(1-\gamma)^2}\left(\left\|B^{h_f}[Q^*_{M^{h_c}}]_M - B[Q^*_{M^{h_c}}]_M\right\|_\infty \right. \\
&\left. + 2\gamma\left\|Q^*_M - [Q^*_{M^{h_c}}]_M\right\|_\infty + \mathrm{Estm}(h_f, D, \delta/3)\right) \\
\leq &\frac{2}{(1-\gamma)^2}\left(3\left\|B^{h_c}[Q^*_{M^{h_c}}]_M - B[Q^*_{M^{h_c}}]_M\right\|_\infty \right. \\
&- \left\|B^{h_f}[Q^*_{M^{h_c}}]_M - B^{h_c}[Q^*_{M^{h_c}}]_M\right\|_\infty \\
&+ \frac{2\gamma}{1-\gamma}\left\|B[Q^*_{M^{h_c}}]_M - [Q^*_{M^{h_c}}]_M\right\|_\infty \\
&\left. + \mathrm{Estm}(h_f, D, \delta/3)\right) \\
\leq &\frac{2}{(1-\gamma)^2}\left(\frac{3-\gamma}{1-\gamma}\left\|B[Q^*_{M^{h_c}}]_M - [Q^*_{M^{h_c}}]_M\right\|_\infty \right. \\
&\left. - \left\|B_D^{h_f}[Q^*_{M^{h_c}}]_M - [Q^*_{M^{h_c}}]_M\right\|_\infty + 2\,\mathrm{Estm}(h_f, D, \delta/3)\right) \\
\leq &\frac{2}{(1-\gamma)^2}\left(\frac{3-\gamma}{1-\gamma}\left\|B[Q^*_{M^{h_c}}]_M - [Q^*_{M^{h_c}}]_M\right\|_\infty \right. \\
&- \left\|B_D^{h_f}[Q^*_{M_D^{h_c}}]_M - [Q^*_{M_D^{h_c}}]_M\right\|_\infty + 2\,\mathrm{Estm}(h_f, D, \delta/3) \\
&\left. + (1+\gamma)\left\|[Q^*_{M_D^{h_c}}]_M - [Q^*_{M^{h_c}}]_M\right\|_\infty\right) \\
\leq &\frac{2}{(1-\gamma)^2}\left(\frac{3-\gamma}{1-\gamma}\left\|B[Q^*_{M^{h_c}}]_M - B^{h_c}[Q^*_{M^{h_c}}]_M\right\|_\infty \right. \\
&\left. + \frac{1+\gamma}{1-\gamma}\,\mathrm{Estm}(h_c, D, \delta/3)\right) \qquad (17) \\
= &\frac{2}{(1-\gamma)^2}\left(\frac{3-\gamma}{1-\gamma}\,\mathrm{Appr}(h_c) + \frac{1+\gamma}{1-\gamma}\,\mathrm{Estm}(h_c, D, \delta/3)\right).
\end{aligned}$$

The derivation is similar to the previous one, with a few small changes. In Equation (16), instead of the Bellman residual we use Lemma 1 to bound the value difference. We also replace Q^*_M with $[Q^*_{M^{h_c}}]_M$, and use Lemma 4 to introduce a term similar to the statistic computed by the algorithm. Then, using the probabilistic guarantees stated at the beginning, we obtain exactly that statistic, and bound it using Equation (4). Finally, $\mathrm{Appr}(h_c)$ appears from Lemma 3 and $\mathrm{Estm}(h_c)$ from the probabilistic guarantees. □

C. Proof Sketch of Theorem 3

We first prove a lemma on Bellman residuals.

Lemma 5. *For any Q-value function $Q : \mathbb{R}^{S\times A}$,*

$$\|BQ - Q\|_\infty \le (1+\gamma)\,\|Q - Q^*_M\|_\infty .$$

Proof.

$$\begin{aligned}\|BQ - Q\|_\infty &= \|BQ - Q^*_M + Q^*_M - Q\|_\infty \\ &\le \|BQ - BQ^*_M\|_\infty + \|Q^*_M - Q\|_\infty \\ &\le \gamma\,\|Q^*_M - Q\|_\infty + \|Q^*_M - Q\|_\infty \\ &= (1+\gamma)\,\|Q^*_M - Q\|_\infty .\end{aligned}$$

So the lemma follows. □

Theorem 3 will a direct corollary of Lemma 6, by noticing that the loss of the selected abstraction can be upper bounded by the LHS of Equation (18).

Lemma 6. *Suppose Assumption 2 holds. Let $\widehat{h_i}$ be the best-so-far abstraction among $h_1, \dots, h_i$ found by Algorithm 2, then for $\delta' = 2\delta/(3|\mathcal{H}|^2)$, the following bound holds w.p. $\ge 1-\delta$: $\forall i = 1, 2, \dots, |\mathcal{H}|$,*

$$\begin{aligned}&\left\|[Q^*_{M^{\widehat{h_i}}}]_M - Q^*_M\right\|_\infty + \frac{1}{1-\gamma}\,\mathrm{Estm}(\widehat{h_i}, D, \delta') \\ &\le poly\big(\frac{1}{1-\gamma}\big)\cdot \min_{h\in\{h_1,\dots,h_i\}} \big(\mathrm{Appr}(h) + \mathrm{Estm}(h, D, \delta')\big).\end{aligned} \tag{18}$$

Proof Sketch. For every pair of possible comparison we require the 3 probabilistic guarantees in the proof of Theorem 2 to hold, hence by union bound we can guarantee that each of them occurs w.p. at least $1-\delta'$. Then, we prove the lemma by induction. For the case of $i = 1$, it holds obviously from Theorem 1, by noticing that the LHS of Lemma 6 is an intermediate step of proving Theorem 1 (up to $2/(1-\gamma)$), and the RHS is consistent with the final bound.

Suppose the induction assumption holds for i, and consider the comparison between $h_c = \widehat{h_i}$ and $h_f = h_{i+1}$. If h_c is selected, we only need to prove that $\left\|[Q^*_{M^{h_c}}]_M - Q^*_M\right\|_\infty + \frac{1}{1-\gamma}\,\mathrm{Estm}(h_c, D, \delta')$ can be bounded by $\mathrm{Appr}(h_f)$ and $\mathrm{Estm}(h_f, D, \delta')$, which is possible by slightly adapting the previous analysis. In particular,

$$\begin{aligned}&\frac{2}{1-\gamma}\left(\|[Q^*_{M^{h_c}}]_M - Q^*_M\|_\infty + \frac{1}{1-\gamma}\,\mathrm{Estm}(h_c, D, \delta')\right) \\ &\le \frac{2}{(1-\gamma)^2}\Big(\left\|B[Q^*_{M^{h_c}}]_M - B^{h_c}[Q^*_{M^{h_c}}]_M\right\|_\infty \\ &\quad + \mathrm{Estm}(h_c, D, \delta')\Big) \\ &\le \frac{2}{(1-\gamma)^2}\Big(\left\|B[Q^*_{M^{h_c}}]_M - B^{h_c}_D[Q^*_{M^{h_c}}]_M\right\|_\infty \\ &\quad + 2\,\mathrm{Estm}(h_c, D, \delta')\Big) \\ &\le \frac{2}{(1-\gamma)^2}\Big(\left\|B[Q^*_{M^{h_c}_D}]_M - B^{h_c}_D[Q^*_{M^{h_c}_D}]_M\right\|_\infty \\ &\quad + 2\gamma\left\|[Q^*_{M^{h_c}_D}]_M - [Q^*_{M^{h_c}}]_M\right\|_\infty + 2\,\mathrm{Estm}(h_c, D, \delta')\Big),\end{aligned}$$

and now we arrive at Equation (12), up to some extra dependence on $\mathrm{Estm}(h_c, D, \delta')$ (which we can always afford), and the difference between δ and δ'. Following the rest part of the previous analysis we will have the desired bound.

If h_f is selected, the beginning part of the previous analysis can be adapted much more easily:

$$\begin{aligned}&\frac{2}{1-\gamma}\left(\left\|[[Q^*_{M^{h_f}}]_M - Q^*_M\right\|_\infty + \frac{1}{1-\gamma}\,\mathrm{Estm}(h_f, D, \delta')\right) \\ &\le \frac{2}{(1-\gamma)^2}\Big(\left\|BQ^*_M - B^{h_f}Q^*_M\right\|_\infty + \mathrm{Estm}(h_f, D, \delta')\Big),\end{aligned}$$

and now we are at Equation (16). This time, however, we cannot follow the previous analysis all the way to the end, as our induction assumption promises nothing for $\mathrm{Appr}(h_c)$ and $\mathrm{Estm}(h_c)$. Instead, we can departure from Equation (17):

$$\begin{aligned}&\text{(17)} \\ &\le \frac{2}{(1-\gamma)^2}\left(\frac{(3-\gamma)(1+\gamma)}{1-\gamma}\,\|[Q^*_{M^{h_c}}]_M - Q^*_M\|_\infty \right. \\ &\quad \left. + \frac{1+\gamma}{1-\gamma}\,\mathrm{Estm}(h_c, D, \delta')\right),\end{aligned}$$

which follows from Lemma 5. Now we can apply our induction assumption, and this shows that the induction assumption holds for $i+1$, so the lemma follows. □