Reinforcement Learning with Polynomial Learning Rate in Parameterized Models

Kirill Dyagilev kirilld@tx.technion.ac.il
Shie Mannor[*] shie@ee.technion.ac.il
Nahum Shimkin shimkin@ee.technion.ac.il
Faculty of Electrical Engineering, Technion, Haifa 32000, Israel

Editor: TBD

Abstract

We consider reinforcement learning in a parameterized setup, where the model is known to belong to a finite set of Markov Decision Processes (MDPs) under the discounted return criterion. We propose an on-line algorithm for learning in such parameterized models, the Parameter Elimination (PEL) algorithm, and analyze its performance in terms of the total mistakes. The algorithm relies on Wald's sequential probability ratio test to eliminate unlikely parameters, and uses an optimistic policy for effective exploration. We establish that with high probability the total mistakes of the algorithm is linear (up to a logarithmic term) in the size of the parameter space, and is independent of the cardinality of the state and action spaces. We further introduce a notion of a decomposable model, which is roughly a system consisting of several independently parameterized subsystems coupled through observed variables. We introduce a version of the PEL algorithm that learns the parameters of each subsystem separately leading to drastic enhancement of the guaranteed learning rate as expressed by the bound on the number of mistakes.

Keywords: Reinforcement Learning, Model–based algorithms, SPRT

1. Introduction

Reinforcement Learning (RL) concerns a learning agent that interacts with a (partially) unknown and possibly stochastic environment, in order to learn optimal control policies (Sutton and Barto, 1998). An obvious goal for efficient learning is fast convergence to the optimal policy. Moreover, in an online setting, the total number of suboptimal decisions made throughout the learning period is of major concern. We shall refer to the latter as the *total mistake count*, and define it more precisely later on.

Several on-line RL algorithms have recently been introduced and shown to provide polynomial bounds (in a PAC sense) on the total mistake count. Such algorithms depend on efficient resolution of the exploration-exploitation tradeoff and include the E^3 algorithm (Kearns and Singh, 2002), the R-max algorithm (Brafman and Tennenholtz, 2002; Kakade, 2003), MBIE (Strehl and Littman, 2005), and Delayed Q-learning (Strehl et al., 2006b, 2009). Since these algorithms make no structural assumptions on the model involved, they essentially rely on the empirical estimation of the model parameters (or value function) for each state and action independently. Consequently, their convergence-rate bounds are at least proportional to

[*] S.Mannor is also with Department of Electrical and Computer Engineering, McGill University, Montreal, Canada H3A-2A7.

the cardinality of the state and action spaces; this may be unacceptable for large problems. Possible approaches to reduce the complexity of learning in large problems include various approximation schemes, such as parametric representations of the value function (Bertsekas and Tsitsiklis, 1996), and state aggregation methods (e.g., Bernstein and Shimkin (2008)). A recent overview may be found in Powell (2007). The effective use of *structural* knowledge regarding the system was demonstrated for factored MDPs in Kearns and Koller (1999).

In this paper we consider the situation where a parameterized model of the system in question is available. The potential simplification offered by such a model in an RL setting can be best demonstrated through a simple example.

Example 1 Consider a discrete time queue, with an input buffer of size K and a single server. The control decision may be whether to admit an arriving customer to the queue, or perhaps idle the server; the specifics are not important here. Without prior knowledge, estimating the system transition structure would require independent sampling at each of the possible B states. However, if we know that the arrival and service processes are geometric with rates that do not depend on the buffer occupancy, then two parameters are sufficient to describe the state dynamics, and these parameters can be estimated by monitoring the arrivals and departures at any system state. We will return to this example in Section 5.

The above example becomes even more distinctive when we consider N queues in parallel (say, with a joint routing controller). Here the state space increases exponentially in N, while the number of parameters increases linearly in N. Obviously, simple-minded learning of the transition probabilities at each state separately makes no sense in this case.

Parameterized control models have been extensively studied in the adaptive control literature (Astrom and Wittenmark, 1995), as well as in the particular context of Markov Decision Processes (MDPs) (Kumar and Varaiya, 1986). However, the results of that line of research are focused mainly on asymptotic convergence, rather than on PAC-like convergence bounds, which are our main concern here.

Our focus in this paper is on parameterized system models with a *finite* parameter space, under the discounted reward criterion. We present an efficient RL algorithm for this problem, called the Parameter Elimination (PEL) algorithm, and show that its total mistake bound grows linearly (up to logarithmic terms) in the size of parameter space, and independently of the size of the state and action spaces.

Essentially, the PEL algorithm is based on eliminating "unlikely" parameters from the list of plausible parameters, J, using Wald's Sequential Probability Ratio Test (SPRT) (Wald, 1952). As for action selection, at every step t an *optimistic* parameter is selected from the set J. This parameter is the one that maximized the (discounted) value function from the current state. The current action is then selected as the optimal one for the optimistic parameter.

The linear dependence of the mistake count on the cardinality of the parameter set is the best that can be attained by *any* learning algorithm, as demonstrated in Example 2. However, as the system becomes more complicated, this number may become unwieldy. In particular, when the overall system is composed of several interconnected subsystems, the parameter cardinality typically increases exponentially in the number of subsystems involved. We address this issue in this paper by considering a *decomposable* model, that consists of several independently parameterized subsystems, which are coupled through observable (input/output) variables. This system is described by a composite parameter vector, with each component of this vector pertaining to a single subsystem. As the size of the composite parameter vector grows exponentially fast in the number of subsystems, the basic PEL algorithm might perform poorly. However, by exploiting the decomposable model structure, the learning rate bounds may be drastically reduced.

Our definition of decomposable models refines the notion of *factored* models introduced in Kearns and Koller (1999) in the following way. In addition to the factored state transition structure, we assume the existence of fully observed coupling variables that determine the interdependencies of the subsystems. Additional observation of the coupling variable allows obtaining separate statistical information on each component of the parameter vector. We provide a variant of the PEL algorithm, the D-PEL algorithm, that is adjusted to decomposable models and has a lower computational and memory complexity that the basic PEL algorithm. We then establish an error bound for the new algorithm that essentially grows *linearly* in the number of subsystems rather than exponentially.

We conclude our paper by presenting simulation results for a simple queueing model with decomposable service and arrival structure. We show, that the number of mistakes the D–PEL algorithm makes is comparable to (or even slightly smaller than) that made by the PEL algorithm. The number of total mistakes of both algorithms turns out to be significantly (times 70) smaller than that of the RTDP-IE algorithm (Strehl et al. (2006a)), which is a state-of-the-art model-based learning algorithm that does not assume any prior knowledge on the model.

The current paper focuses on the case of a *finite* parameter set. While this case is of interest on its own, it may also serve as an intermediate step for treating the continuous parameter case via discretization. A detailed treatment of this approach is beyond the scope of the present paper and is presented in Dyagilev (2009).

The rest of the paper is organized as follows. In Section 2.1 we present the model along with some definitions and notations. Section 2.2 defines the main performance metrics considered in this paper. In Section 3 we present the PEL algorithm and provide our main performance bounds for this algorithm. Section 4 is devoted to the proof of these results and discusses several aspects of the obtained error bound. In Section 5 we introduce the concept of decomposable models and describe the D-PEL algorithm. In Section 6 we show simulation results. We conclude with a summary and discussion of future work in Section 7.

2. Preliminaries

In this section we provide a rigorous definition of the MDP and quote several of its basic properties. We further introduce algorithm performance metrics that is used in this paper.

2.1 Model Formulation

An MDP M is specified by a five-tuple $\langle S, A, R, p, \eta\rangle$, where S is a finite state space, A is a finite action space, R is a finite reward set, $p : S \times A \to \Delta(S)$ is the transition kernel and $\eta : S \times A \to \Delta(R)$ is the reward distribution function. Here $\Delta(S)$ denotes the set of probability vectors over the set S, and similarly for $\Delta(R)$. Given that at the time step t the state is $s_t \in S$ and the action is $a_t \in A$, the agent receives a reward $r_t \in R$ generated via probability distribution $\eta(\cdot|s_t, a_t)$ and moves to state $s_{t+1} \in S$ with probability $p(s_{t+1}|s_t, a_t)$.

The observed history until time t is the sequence $h_t \overset{\triangle}{=} \{s_0, a_0, r_0, ..., s_{t-1}, a_{t-1},\ r_{t-1}, s_t\}$. A (deterministic) decision rule is a mapping from history to action, namely $\pi_t : H_t \to A$, where $H_t = (S \times A \times R)^t \times S$. A policy $\mathcal{A}$ is a collection of decision rules $\{\pi_t\}_{t=0}^{\infty}$ so that $a_t = \pi_t(h_t)$. Note that a (deterministic) learning algorithm is such a policy. Given an initial state s, the policy $\mathcal{A}$ induces a stochastic process $(s_t, a_t, r_t)_{t=0}^{\infty}$ with probability measure $\mathbb{P}^{\mathcal{A},s}\cdot$. The expectation operator corresponding to this measure is denoted by $\mathbb{E}^{\mathcal{A},s}$.

Let $V^{\mathcal{A}}(s) \overset{\triangle}{=} \mathbb{E}^{\mathcal{A},s}\left\{\sum_{t=0}^{\infty} \gamma^t r_t\right\}$ denote the discounted return for policy $\mathcal{A}$ from state s. Here $0 < \gamma < 1$ is the discount factor, which we fix from now on. We refer to $V^{\mathcal{A}}(s)$ as the value function for policy $\mathcal{A}$. A policy $\mathcal{A} = \{\pi_t\}_{t=0}^{\infty}$ is called stationary if $\pi_t = \pi$ for all t, where $\pi : S \to A$ is a function of the current state only. It is well known (e.g., Puterman (1994)) that there exists a deterministic stationary policy π^* that is optimal in sense that $V^{\pi^*}(s) \geq V^{\mathcal{A}}(s)$ for any state s and any policy $\mathcal{A}$. Denote the corresponding optimal value function by $V^*(\cdot)$. Further define the action-value function (or Q-function) for state-action pair (s,a) as $Q^*(s,a) = \bar{r}(s,a) + \gamma \sum_{s' \in S} p(s'|s,a)V^*(s')$, where $\bar{r}(s,a) \overset{\triangle}{=} \sum_{r \in R} r\eta(r|s,a)$. The following equality, known as Bellman equation, holds for any stationary policy π and state $s \in S$:

$$V^{\pi}(s) = \bar{r}(s,\pi(s)) + \gamma \sum_{s' \in S} V^{\pi}(s')p(s'|s,\pi(s)),$$

while the optimal value function of policy $\pi^*(s)$ satisfies

$$V^*(s) = \max_a Q^*(s,a) = Q^*(s,\pi^*(s)).$$

Let $R_{\max}$ denote an upper bound on the expected one-step reward, so that $\bar{r}_\theta(s,a) \leq R_{\max}$ for all $\theta \in \Theta$, $s \in S$ and $a \in A$. Let $R_{\max}$ denote an upper bound on the one-step reward, that is $r(s,a) \leq R_{\max}$ for all $s \in S$ and $a \in A$.

In this paper we assume that there is a known family $\{M_\theta\}_{\theta \in \Theta}$ of parameterized models, where $\Theta = \{\theta_0, \theta_1, ..., \theta_{|\Theta|-1}\}$ is a finite set of representative parameter values. All models in the given family share the same action, reward and state spaces, while their transition and reward probabilities depend on the parameter $\theta \in \Theta$, i.e., $M_\theta =< S, A, R, p_\theta, \eta_\theta >$. For each MDP M_θ we denote by π^*_θ, V^*_θ and Q_θ an optimal stationary policy, the optimal value function and the Q-function, respectively. In case the optimal policy is not unique, we henceforth fix one (arbitrary) selection. For brevity of exposition, we define $\hat{\theta}$ to be the *true parameter*, namely, the actual model M is given by $M =< S, A, R, p_{\hat{\theta}}, \eta_{\hat{\theta}} >\equiv M_{\hat{\theta}}$. In the discrete parameter case we assume that $\hat{\theta} \in \Theta$. In the mismatched case the true parameter $\hat{\theta}$ need not belong to Θ, however, there exists a representative $\theta \in \Theta$ so that models M_θ and $M_{\hat{\theta}}$ are sufficiently close in their properties. We denote by $\pi^*_{\hat{\theta}}$, $V^*_{\hat{\theta}}$ and $Q_{\hat{\theta}}$ an optimal stationary policy, the optimal value function and the Q-function for the true model $M_{\hat{\theta}}$, respectively.

2.2 Performance Metrics

An effective measure of on-line learning efficiency is the number of time steps the algorithm prescribes sub-optimal action. Recall that an optimal action $a^* = \pi^*(s)$ in state s satisfies $Q^*(s,a^*) = V^*(s)$. Hence the difference $V^*(s) - Q^*(s,a)$ quantifies the effect of taking a single suboptimal action a at state s, and thereafter proceeding optimally. The action mistake count is defined as follows:

Definition 1 *We define the* action mistake count *(AMC) as a total number of ϵ-suboptimal state-action pairs visited by the algorithm during its operation, namely,*

$$AMC(\epsilon) \overset{\triangle}{=} \sum_{t=0}^{\infty} \mathbb{I}\left\{Q^*(s_t, a_t) < V^*(s_t) - \epsilon\right\}.$$

Note that for ϵ small enough $AMC(\epsilon) = AMC(0)$ (due to the finiteness of the state and action spaces), so that all non-optimal actions are counted.

We further introduce a more elaborated performance criterion relies on a quantification of "sub-optimality" of a policy rather than a single action. Denote by $\mathcal{A}$ the policy of the learning algorithm. Let h_τ be the observed history up to time τ, and denote by $V^{\mathcal{A}}(h_\tau) \overset{\triangle}{=} \mathbb{E}^{\mathcal{A},s_0}\left\{\sum_{j=\tau}^{\infty} \gamma^{j-\tau} r_j \middle| h_\tau\right\}$ the value of the policy $\mathcal{A}$ starting from time τ. The policy mistake count is defined as follows:

Definition 2 *Let ϵ be a positive number. The time step t in said to be an ϵ-**suboptimal step** if $V^{\mathcal{A}}(h_t) < V^*(s_t) - \epsilon$. Equivalently, we say that the learning agent follows an ϵ-**suboptimal policy** at time t. The **policy-mistake count** (PMC) of a learning algorithm is defined as $PMC(\epsilon) \overset{\triangle}{=} \sum\limits_{t=0}^{\infty} \mathbb{I}\left\{V^{\mathcal{A}}(h_t) < V^*(s_t) - \epsilon\right\}$.*

The PMC criterion was suggested by Kakade (2003) and originally called the "sample complexity of exploration".

It is easily verified that for any $\epsilon > 0$ and learning algorithm AMC is dominated by PMC (see Lemma 2.3 in Dyagilev (2009)), i.e., $AMC(\epsilon) \leq PMC(\epsilon)$. It follows that any upper bound on the PMC also applies to the AMC. For this reason we shall focus in the following on PMC alone. We now define the corresponding notion of a PPAC algorithm.

Definition 3 *A learning algorithm A is called polynomial **PMC-PAC** (or just PPAC) if, for any positive ϵ and δ, its policy-mistake count (and hence action-mistake count) is polynomial in $(\epsilon^{-1}, \delta^{-1}, (1-\gamma)^{-1}, |\Theta|)$ with probability of at least $(1-\delta)$.*

3. The Parameter Elimination Algorithm

In our discrete–parameter setting, the learning problem may be reduced to the identification of the true parameter or, at least, a parameter that leads to an ϵ-optimal control policy for the true model. Equivalently, one may try to eliminate all other parameters from the set of optional parameters.

Define the log-likelihood function of the observation $(s_{t-1}, a_{t-1}, r_{t-1}, s_t)$ at time step t as

$$l_t(\theta) = \log p_\theta(s_t | s_{t-1}, a_{t-1}) + \log \eta_\theta(r_{t-1} | s_{t-1}, a_{t-1}). \tag{1}$$

The *cumulative* log-likelihood is then $G_t(\theta) = \sum_{i=1}^{t} l_t(\theta)$.

The PEL algorithm proceeds as follows (see Algorithm 1 for details). As an input, the algorithm requires the finite family of possible MDPs $\{M_\theta\}_{\theta\in\Theta}$, with common state, reward and action spaces. The value function $V_\theta^*(\cdot)$ and the optimal policy $\pi_\theta^*(\cdot)$ for each model can be calculated using one of the standard algorithms, i.e., value iteration, policy iteration or linear programming (see Puterman (1994)). An allowed probability of error δ is also provided as input.

Algorithm 1 Parameter ELimination

```
Input: $\{M_\theta\}_{\theta\in\Theta}$ – the finite family of possible MDPs, $\delta$ – an allowed probability of error.
Initialize: Initialize the list of plausible parameter values to $J_0 = \Theta$. Initialize the array of cumulative log-likelihood to $G_0(\theta) = 0$ for all $\theta \in \Theta$.
For $t = 0, 1, ...$ do
1. Stopping condition: If $J_t$ is a singleton, namely $J_t = \{\theta\}$, then use the corresponding policy $\pi^*_\theta$ indefinitely and skip items (2)-(5) below.
2. Find an optimistic parameter: Select a parameter value that maximizes the value function among plausible parameter values: $\theta(t) = \arg\max_{\theta\in J_t} V^*_\theta(s_t)$.
3. Act: Execute the action according to the optimal policy for the optimistic parameter: $a_t = \pi^*_{\theta(t)}(s_t)$.
4. Update: Observe the reward $r_t$ and the next state $s_{t+1}$. Update for all $\theta \in J_t$: $G_{t+1}(\theta) = G_t(\theta) + l_{t+1}(\theta)$ where $l_{t+1}$ is defined in (1).
5. Eliminate: Set $J_{t+1} = J_t$ and do:
    a. For all $\theta \in J_{t+1}$ so that $G_{t+1}(\theta) = -\infty$, let $J_{t+1} = J_{t+1} \setminus \{\theta\}$.
    b. Find the most likely parameter in the plausible set $\hat{\theta} = \arg\max_{\theta\in J_{t+1}} G_{t+1}(\theta)$.
    c. For all $\theta \in J_{t+1}$ so that $G_{t+1}(\hat{\theta}) - G_{t+1}(\theta) > \log\left[\frac{3(|\Theta|-1)}{\delta}\right]$, let $J_{t+1} = J_{t+1} \setminus \{\theta\}$.
```

The algorithm maintains a list of plausible parameters J_t throughout its execution. Initially, all parameter values are considered plausible and then they are eliminated one by one. The elimination step is based on the Sequential Probability Ratio Test (SPRT), namely, comparing the log-likelihood ratio $G_t(\theta_i) - G_t(\theta_j)$ to a given threshold $G_{th} > 0$. If at time step t there exist parameters $\theta_i, \theta_j \in J_t$ so that $G(\theta_i) - G(\theta_j) > G_{th}$ then θ_j is eliminated. Equivalently, we first find $\hat{\theta}$, the most likely parameter in the set J_t, and then compare the likelihood of all other plausible parameters to $G(\hat{\theta})$. As the error probability of each elimination can be upper bounded by $e^{-G_{th}}$, the selection of $G_{th} = \log[3(|\Theta| - 1)/\delta]$ yields cumulative error probability of all eliminations less than $\frac{\delta}{3}$ (see Dyagilev et al. (2009) for details).

The exploration-exploitation tradeoff is addressed using the so-called "optimism in face of uncertainty" principle. At each time step t, the PEL algorithm selects an "optimistic" action in the following sense. First, the algorithm selects the parameter $\theta(t) \in J_t$ that maximizes the value function $V^*_\theta(s_t)$ for the current state s_t. The selected action is then the optimal one given $\theta(t)$, i.e., $a_t = \pi^*_{\theta(t)}(s_t)$. We note that the selected action may correspond to a different parameter θ at each state, even if the set J_t does not change.

The high probability bound on the mistake count of the PEL algorithm is given by the following theorem:

Theorem 4 *Consider the PEL algorithm with parameter $0 < \epsilon < \frac{R_{\max}}{(1-\gamma)}$ and $0 < \delta < 1$. With probability of at least $1 - \delta$, PEL's policy-mistake count is upper bounded by*

$$PMC(\epsilon) \le L(|\Theta|, \epsilon, \delta, \gamma)\,|\Theta|\,\frac{R^3_{\max}}{\epsilon^3(1-\gamma)^6}\log\left(\frac{3\,|\Theta|}{\delta}\right), \tag{2}$$

where $L(|\Theta|, \epsilon, \delta, \gamma) = 1000 \log\frac{4R_{\max}}{\epsilon(1-\gamma)}$.

This theorem implies that the PEL algorithm is PPAC in terms of the total mistake bound, and its PMC is linear (up to the logarithmic term $L(\cdot)$) in the size of the parameter set. Note that the bound is independent of the cardinality of the state and action spaces.

Remark 5 As in Strehl and Littman (2005) and Strehl et al. (2009), Theorem 19 provides a bound on the total number of mistakes the algorithm makes rather than on the number of time steps until convergence to the optimal policy. Hence, notion of *mixing time*, which is roughly the number of time steps it takes to reach some recurrent state of the MDP (e.g., see Kearns and Singh (2002)), are irrelevant for our analysis.

4. Proof of the Main Result

An outline of the proof of Theorem 4 is as follows. We begin in Section 4.1 by introducing an optimistic auxiliary model that will prove useful later on. In Section 4.2 we define *informative state-action pairs* (Definition 9) that are roughly state-action pairs that distinguish the true MDP and the auxiliary model. We next show in Lemma 10 that there is a positive probability to reach an informative state-action pair within a finite time interval following an ϵ-suboptimal time step (Definition 2). Moreover, Lemma 11 (Section 4.3) implies that the number of ϵ-suboptimal steps encountered can be bounded with high probability in terms of number of actual visits to informative state-action pairs. Hence, once we show that the number of visits to informative state-action pairs is bounded, we can conclude that the policy-mistake count is bounded as well. To show the former, we bound in Section 4.5 the stopping time of the SPRT test (for any fixed parameter $\theta \neq \theta_0$) using a non-decreasing measure of accumulated statistical information related to Bhattacharyya's information coefficient. In Section 4.6 we show that each visit to an informative state-action pair adds some strictly positive amount of information to one parameter at least. Hence the number of visits needed for SPRT to trigger is bounded. Using the pigeon-hole principle, we obtain that the number of visits to an informative state action pairs until convergence to an ϵ-optimal policy is also bounded, thus concluding the proof.

Note that from this point on all the probabilities and expectations refer to the stochastic process induced by the PEL algorithm on the actual MDP M_{θ_0}, unless mentioned otherwise.

4.1 An Auxiliary Model

Consider a *fixed* subset of parameters $J \subseteq \Theta$. For every $s \in S$, define the *optimistic parameter* in J as $\theta(J,s) = \arg\max_{\theta \in J} V^*_\theta(s)$ (with ties decided arbitrarily). Define an auxiliary MDP $M_J = \langle S, A, R, p_J, \eta_J \rangle$, where $p_J(s'|s,a) = p_{\theta(J,s)}(s'|s,a)$ and $\eta_J(r'|s,a) = \eta_{\theta(J,s)}(r'|s,a)$. Further, define the following stationary policy: $\pi_J(s) = \pi^*_{\theta(J,s)}(s)$. This policy picks at each state the optimal action according to the parameter $\theta(J,s)$ that is optimistic for that state. (In the context of the PEL algorithm, it is evident that as long as the set J_t is equal to J, the algorithm follows this stationary policy.) Denote the value function of the MDP M_J under the policy π_J as $V_J^{\pi_J}$. For notational convenience we use the abbreviated notation V_J. Then the auxiliary model is optimistic in the following sense (see Appendix A for the proof):

Lemma 6 *For any $s \in S$ and $\theta \in J$ it holds that*[1] $V_J(s) \geq V^*_\theta(s)$.

[1] Note that the auxiliary model M_J need not be in the family $\{M_\theta\}_{\theta\in\Theta}$. Hence, it may even hold that $V_J(s) > V^*_\theta(s)$ for every $\theta \in \Theta$ and $s \in S$.

4.2 Implicit Explore or Exploit

We next prove that the PEL algorithm implicitly provides a tradeoff between exploration and exploitation. In other words, the agent either follows an ϵ-optimal policy or otherwise gains some information with a positive probability.

The proof is partially based on results from Strehl and Littman (2005) and Kearns and Singh (2002). For a stationary policy π denote the H-step value function by $V^{\pi}(s,H) \overset{\triangle}{=} \mathbb{E}^{\pi,s}\left\{\sum_{t=0}^{H-1}\gamma^t r_t\right\}$. The first lemma addresses the sensitivity of the value function to the time horizon.

Lemma 7 *If $H \geq \frac{1}{1-\gamma}\log\frac{R_{\max}}{\epsilon(1-\gamma)}$ then $|V^{\pi}(s,H)-V^{\pi}(s)| \leq \epsilon$ for all policies π and states s.*

Proof The result follows easily by bounding the tail of sum of rewards in the definition of the value function; see, e.g., Lemma 2 in Kearns and Singh (2002). ■

In the following we use $T_{\text{eff}} = \frac{1}{1-\gamma}\log\frac{4R_{\max}}{\epsilon(1-\gamma)}$ as an effective horizon length, beyond which the effect on the discounted return is smaller than $\epsilon/4$.

The following lemma bounds the sensitivity of the discounted reward function to perturbations in the transition and reward probabilities. For two probability distributions p and q on a finite set A, we use the l_1 norm to measure their separation: $\|p(\cdot)-q(\cdot)\|_1 = \sum_{a\in A}|p(a)-q(a)|$.

Lemma 8 *Let $M_1 =< S,A,R,p_1,\eta_1 >$ and $M_2 =< S,A,R,p_2,\eta_2 >$ be two MDPs with non-negative rewards bounded by $R_{\max}$. Let π be some stationary policy and let ϵ be a positive number. If $\|\eta_1(\cdot|s,a)-\eta_2(\cdot|s,a)\|_1 \leq \frac{\epsilon(1-\gamma)^2}{R_{\max}}$ and $\|p_1(\cdot|s,a)-p_2(\cdot|s,a)\|_1 \leq \frac{\epsilon(1-\gamma)^2}{R_{\max}}$ for all states s and actions a, then $\max_{s\in S}\left|V^{\pi}_{M_1}(s)-V^{\pi}_{M_2}(s)\right| \leq \epsilon$.*

Proof The lemma follows from Lemma 4 in Strehl and Littman (2005), after noting that

$$|\bar{r}_1(s,a)-\bar{r}_2(s,a)| \leq R_{\max}\|\eta_1(\cdot|s,a)-\eta_2(\cdot|s,a)\|_1 .$$

■

To state the central result of this subsection, we define informative state-action pairs as those pairs for which either the state transition or the reward distribution are distinct under the true and optimistic models. More precisely:

Definition 9 *Recall that θ_0 is the true parameter. Let $\theta(J,s)$ be defined as in Section 4.1. For $t \geq 0$, let K_t be the set of state-action pairs (s,a) for which $\left\|\eta_{\theta(J_t,s)}(\cdot|s,a)-\eta_{\theta_0}(\cdot|s,a)\right\|_1 \leq \frac{\epsilon(1-\gamma)^2}{4R_{\max}}$, and $\left\|p_{\theta(J_t,s)}(\cdot|s,a)-p_{\theta_0}(\cdot|s,a)\right\|_1 \leq \frac{\epsilon(1-\gamma)^2}{4R_{\max}}$. We say that the PEL algorithm visited an **informative state-action pair** at time t, if $(s_t,a_t) \notin K_t$.*

The following proposition asserts that occurrence of an ϵ-suboptimal step leads to an explorative interval, where an informative state-action pair is visited with probability of at least $\frac{\epsilon(1-\gamma)}{2R_{\max}}$. Recalling the definition of an ϵ-suboptimal time step in Definition 2, let

$$E_1(t) \overset{\triangle}{=} \{\theta_0 \in J_t\} \cap \{V^{\mathcal{A}_t}(h_t) < V^*_{\theta_0}(s_t)-\epsilon\},\ t \geq 0 \tag{3}$$

denote the event that the action at time step t is ϵ-suboptimal and the true parameter was not eliminated before time t. Let

$$E_2(t) \overset{\triangle}{=} \{(s_{t-1}, a_{t-1}) \notin K_{t-1}\} \cup \{J_t \neq J_{t-1}\},\ t \geq 1 \tag{4}$$

be the event that at time step $(t-1)$ either an informative state-action pair was visited or some parameter was eliminated from the set J_{t-1} of plausible parameters at time t. Denote by $E_3(t) \overset{\triangle}{=} \bigcup_{\tau=t+1}^{t+T_{\text{eff}}} E_2(\tau)$ the event that the informative event $E_2(\tau)$ occurred for τ between $(t+1)$ and $(t+T_{\text{eff}})$. Let $\mathcal{F}_t \overset{\triangle}{=} \sigma\{h_t\}$ be the sigma algebra of the history sequence until time step t. Then $E_1(t), E_2(t) \in \mathcal{F}_t$, while $E_3(t) \in \mathcal{F}_{t+T_{\text{eff}}}$.

Proposition 10 *For every t and history h_t that satisfies $E_1(t)$, $\mathbb{P}^{\mathcal{A},s_0}\left\{E_3(t) \middle| h_t\right\} > \frac{\epsilon(1-\gamma)}{2R_{\max}}$.*

Proof See Appendix B for the proof. ■

4.3 A Discovery Lemma

Proposition 10 shows that in the T_{eff} steps following an ϵ-suboptimal step there is a probability of at least $\frac{\epsilon(1-\gamma)}{2R_{\max}}$ to reach some informative state-action pair or eliminate some parameter from J_t. Based on that, Lemma 11 below bounds the number of ϵ-suboptimal steps in terms of the number of actual visits to informative state-action pairs and parameter eliminations.

Let K_t be as in Definition 9 and let N_2 be a positive integer. Recall the definitions of $E_1(t)$, $E_2(t)$, $E_3(t)$ and $\mathcal{F}_t$ from the previous section.

Lemma 11 *For any positive integer N_2, let $T_2(N_2)$ be the time step on which the event $E_2(t)$ occurred for the N_2-th time, namely,*

$$T_2(N_2) = \inf\left\{ n \geq 1 \,\middle|\, \sum_{k=1}^{n} \mathbb{I}\left\{E_2(k)\right\} = N_2 \right\} \tag{5}$$

(with $T_2(N_2) = \infty$ is such n does not exist). Then, for all $\epsilon > 0$ and $0 < \delta < 1$, $\mathbb{P}^{\mathcal{A},s_0}\left\{\sum_{k=0}^{T_2(N_2)} \mathbb{I}\left\{E_1(k)\right\} \leq N_1\right\} \geq 1-\delta$, where $N_1 \overset{\triangle}{=} \frac{4R_{\max}T_{eff}}{\epsilon(1-\gamma)}\left[N_2 + \frac{8R_{\max}}{\epsilon(1-\gamma)} \log \frac{T_{eff}}{\delta_3}\right]$.

Proof See Appendix C for the proof. ■

4.4 Sequential Hypothesis Testing

The sequential hypothesis test we use in our algorithm was originated by Wald (1952) and is defined in the following way. Consider a discrete-time stochastic process $\{x_t\}_{t=0}^{\infty}$ taking values in a finite set S. Denote by $x_0^n = \{x_0, ..., x_n\}$ the observations obtained by time n. Let the probability of such observations under hypothesis H_0 be denoted by $p_0(x_0^n)$, and under H_1 by $p_1(x_0^n)$. Note that the discussion here is not limited to Markov processes.

Definition 12 *For any $0 < \delta < 1$ define the stopping time*

$$N^W(\delta) = \inf\left\{ n \geq 1 : \left| \log \frac{p_1(x_0^n)}{p_0(x_0^n)} \right| \geq -\log \delta \right\},$$

and the decision rule $d^W(\delta)$ that chooses upon stopping a more likely hypothesis. Then the pair $\left(N^W(\delta), d^W(\delta)\right)$ is called the Sequential Probability Ratio Test (SPRT).

It was shown by Wald (1952) that the error probability of the SPRT is bounded by δ:

Theorem 13 (Wald) $\mathbb{P}\left\{ d^W(\delta) = H_0 \middle| H_1 \right\} \le \delta$ *and* $\mathbb{P}\left\{ d^W(\delta) = H_1 \middle| H_0 \right\} \le \delta$.

We next establish a useful bound on the stopping time of SPRT, using an auxiliary stopping time for the same process based on the Bhattacharyya coefficient rather than the likelihood ratio. We begin by defining the Bhattacharyya coefficient (Kailath, 1967).

Definition 14 *Let p and q be probability distributions on a finite set S. Then the* ***Bhattacharyya coefficient*** *is $\rho \triangleq \sum\limits_{s' \in S} p^{1/2}(s') q^{1/2}(s')$.*

Note that $\rho \le 1$ by the Cauchy-Schwarz inequality. The Bhattacharyya distance (or information) is defined as $-\log \rho$. This metric is related to the l_1-norm of $(p-q)$ in the following way (see Appendix D for the complete proof):

Lemma 15 *Let p and q be probability distributions on a finite set S. Then,*
$-\log \rho \ge \frac{1}{8} \|p-q\|_1^2$.

Definition 16 *Consider the same processes and hypotheses as in Definition 12. Denote by $\rho(x_0^n) = \sum\limits_{x' \in S} p_0^{1/2}(x'|x_0^n) p_1^{1/2}(x'|x_0^n)$ the Bhattacharyya coefficient between $p_0(\cdot|x_0^n)$ and $p_1(\cdot|x_0^n)$. Then the* ***Bhattacharyya stopping time*** *with parameter $0 < \delta < 1$ is defined as:*

$$N^B(\delta) = \inf \left\{ n \ge 1 \left| \prod_{t=0}^{n-1} \rho(x_0^t) \le \delta, \text{ or } p_1(x_n | x_0^{n-1}) = 0 \right. \right\}. \tag{6}$$

We note that the stopping condition $\prod_{t=0}^{n-1} \rho(x_0^t) \le \delta$ can be written as $R_n \triangleq -\sum_{t=0}^{n-1} \log \rho(x_0^t) \ge -\log \delta$, where R_n is the cumulative Bhattacharyya distance (or total Bhattacharyya information).

While our algorithm uses the Wald test, the Bhattacharyya stopping time will be more handy for analysis as R_n is a non-decreasing sequence. The following proposition relates these two stopping times (see Appendix E for the proof).

Proposition 17 *For $0 < \delta < 1$, the inequality $\mathbb{P}\left\{ N^W(\delta) > N^B(\delta^{3/2}) \right\} \le \delta$ holds both under H_0 and H_1.*

4.5 Information Count Lemma

Consider the PEL algorithm applied to the true MDP M_{θ_0} and consider hypothesis testing between MDPs M_{θ_0} and M_θ for $\theta \ne \theta_0$. Denote by $R_t(\theta)$ the total Bhattacharyya information of the observed history h_t with respect to hypotheses M_{θ_0} and M_θ. Denote by $N^B(\theta, \delta)$ and $N^W(\theta, \delta)$ the corresponding Bhattacharyya stopping time (see Definition 16) and the corresponding SPRT stopping time (see Definition 12).

The following lemma assesses the maximal number of visits to informative state-action pairs until all parameter $\theta \ne \theta_0$ are eliminated.

Lemma 18 *Let $\delta' \triangleq \delta / \left[3(|\Theta|-1)\right]$. Consider a history $h_\infty = \{s_t, a_t, r_t\}_{t=0}^{\infty}$ so that the following two conditions are satisfied:*
(1) The true parameter θ_0 is not eliminated from the plausible list J during the execution of the PEL algorithm; and
(2) $N^W(\theta, \delta') \le N^B\left(\theta, (\delta')^{3/2}\right)$ for all $\theta \ne \theta_0$, i.e., the relation between Bhattacharyya stopping time and SPRT stopping time is as indicated by Lemma 17.
Then the total number of visits to informative state-action pairs before all $\theta \ne \theta_0$ are eliminated is upper bounded by $(|\Theta|-1)\log((1/\delta')^{3/2})/B_0$, where $B_0 \triangleq \frac{1}{8}\left(\epsilon(1-\gamma)^2/(4R_{\max})\right)^2$.

Proof As mentioned in Section 3, the elimination step of the algorithm can be interpreted as an SPRT between any pair of parameter in J_t, with the threshold of δ'. Thus, the elimination time of any $\theta \ne \theta_0$ can be bounded from above by $N^W(\theta, \delta')$, which by condition (2) is dominated by $N^B\left(\theta, (\delta')^{3/2}\right)$. In what follows we bound the total number of visits to informative state-action pairs before time $\max_{\theta \ne \theta_0} N^B\left(\theta, (\delta')^{3/2}\right)$ stops, thus bounding the number of these visits until all $\theta \ne \theta_0$ are eliminated from the plausible set J.

Let t be a time step on which an informative state-action pair (s_t, a_t) is visited (see Definition 9). Let us assess the Bhattacharyya distance $(-\log \rho_t)$ between the joint distribution of (r_t, s_{t+1}) under the true model M_{θ_0} and the auxiliary model M_J. Evidently,

$$-\log \rho_t = -\log\left[\sum_{s\in S} p_{\theta(t)}^{1/2}(s|s_t,a_t)p_{\theta_0}^{1/2}(s|s_t,a_t)\right] - \log\left[\sum_{r\in S} \eta_{\theta(t)}^{1/2}(r|s_t,a_t)\eta_{\theta_0}^{1/2}(r|s_t,a_t)\right],$$

where $\theta(t)$ is the optimistic parameter at time t (see Algorithm 1). Since $(s_t, a_t) \notin K_t$, it follows by Lemma 15 that $-\log \rho_t > \frac{1}{8}\left(\epsilon(1-\gamma)^2/(4R_{\max})\right)^2 \triangleq B_0$. Hence, each visit to an informative state-action pair $(s_t, a_t) \notin K_t$ increases $R_t(\theta)$ by at least B_0 for at least one $\theta \in J_t$. As the sequence $R_t(\theta)$ is non-decreasing, the total number of such increments until the time $N^B(\theta, (\delta')^{3/2})$ stops is upper bounded by $\log((1/\delta')^{3/2})/B_0$.

By the pigeon-hole principle it will take less than $(|\Theta|-1)\log((1/\delta')^{3/2})/B_0$ until all times $N^B(\theta, \delta')$ for $\theta \ne \theta$ stop. ■

4.6 Proof of Theorem 4

Consider the PEL algorithm applied to the true MDP M_{θ_0}. The proof proceeds through the following steps. In steps 1-3 we define three "unwanted" events: the event E_4 on which the true parameter θ_0 is eliminated from the plausible parameter set J_t at some point; the event E_5 on which (essentially) there is insufficient number of visits to informative state-action pairs despite a large number of "sub-optimal" steps; and the event E_6 on which a sufficient amount of Bhattacharyya information does not lead to parameter elimination in the SPRT test. We show that the probability of each is bounded by $\delta/3$. In step 4 and step 5 the required upper bound on the PMC is shown to hold on the complement of $E_4 \cup E_5 \cup E_6$. In step 6 we combine the above to conclude the required result.

Step 1: Let $E_4 \triangleq \{\theta_0 \notin \cap_{t=1}^{\infty} J_t\}$ be the event that the actual parameter is eliminated from the set J_t of plausible parameters at some point. As mentioned, the elimination step of the algorithm can be interpreted as a SPRT between any pair of parameter in J_t, with the

threshold of $\delta' \triangleq \frac{\delta}{3(|\Theta|-1)}$. From Theorem 13 we obtain that the probability of eliminating θ_0 due to any other fixed parameter is less than δ'. Therefore, by union bound the total probability of eliminating θ_0 is less then $(|\Theta|-1)\delta'$, namely, $\mathbb{P}^{\mathcal{A},s_0}\left\{E_4\right\} \leq (|\Theta|-1)\delta' = \delta/3$.
Step 2: Recall the definition of $E_1(t)$ and T_2 from (3) and (5). Let

$$E_5 \triangleq \left\{ \sum_{t=1}^{T_2(N_2)} \mathbb{I}\left\{E_1(t)\right\} > N_1 \right\}$$

be the event that the event $E_1(t)$ was encountered more than N_1 times before the N_2-th occurrence of the event $E_2(t)$. Here,

$$N_2 \triangleq 12(|\Theta|-1)\left(\frac{4R_{\max}}{\epsilon(1-\gamma)^2}\right)^2 \log(\frac{1}{\delta'}) + (|\Theta|-1)$$

(this selection is explained in step 4) and N_1 is selected as in Lemma 11 with $\delta = \delta/3$, namely,

$$N_1 \triangleq \frac{4R_{\max}T_{\text{eff}}}{\epsilon(1-\gamma)}\left[N_2 + \frac{8R_{\max}}{\epsilon(1-\gamma)}\log\frac{3T_{\text{eff}}}{\delta}\right].$$

Then, Lemma 11 implies (for any N_2 and in particular for the one above), $\mathbb{P}^{\mathcal{A},s_0}\left\{E_5\right\} \leq \delta/3$.
Step 3: Consider hypothesis testing between MDPs M_{θ_0} and M_θ for $\theta \neq \theta_0$. Denote by $N^W(\theta,\delta)$, $R_t(\theta)$ and $N^B(\theta,\delta)$ the corresponding SPRT stopping time, the total Bhattacharyya information and the Bhattacharyya stopping time (see Definitions 12 and 16). Let E_6 be the event on which $N^W(\theta,\delta') > N^B\left(\theta,(\delta')^{3/2}\right)$ holds for some $\theta \neq \theta_0$ (i.e., the relation between Bhattacharyya stopping time and SPRT stopping time defined in Lemma 17 is violated). Using Proposition 17 and the union bound we conclude that $\mathbb{P}^{\mathcal{A},s_0}\left\{E_6\right\} \leq (|\Theta|-1)\delta' = \delta/3$.
Step 4: Consider a realization $h_\infty = \{s_t,a_t,r_t\}_{t=0}^{\infty} \in E_4^c \cap E_5^c \cap E_6^c$, where A^c stands for complement event for the event A. Noting the definition of the event $E_2(t)$ in (4), recall that $E_2(t)$ occurs if an informative state-action pair was visited at time $(t-1)$ or a parameter was eliminated from J_{t-1}. Hence,

$$\sum_{t=1}^{\infty}\mathbb{I}\left\{E_2(t)\right\} \leq \sum_{t=1}^{\infty}\mathbb{I}\left\{(s_{t-1},a_{t-1}) \notin K_{t-1}\right\} + \sum_{t=1}^{\infty}\mathbb{I}\left\{J_t \neq J_{t-1}\right\}.$$

We bound the first term using Lemma 18 yielding

$$\sum_{t=1}^{\infty}\mathbb{I}\left\{E_2(t)\right\} \leq (|\Theta|-1)\frac{\log((1/\delta')^{3/2})}{B_0} + (|\Theta|-1) \equiv N_2.$$

Step 5: Let T_2, N_2 be as in Step 2. For h_∞ as before we argue that $PMC(\epsilon) \leq N_1$. Since

$h_\infty \in E_5^c$,

$$\begin{aligned}
N_1 &\geq \sum_{t=0}^{T_2(N_2)} \mathbb{I}\left\{E_1(t)\right\} \\
&= \left[\sum_{t=0}^{\infty} \mathbb{I}\left\{E_1(t)\right\}\right] \mathbb{I}\left\{T_2(N_2) = \infty\right\} + \left[\sum_{t=0}^{\infty} \mathbb{I}\left\{E_1(t)\right\}\right] \mathbb{I}\left\{T_2(N_2) < \infty\right\} \\
&- \left[\sum_{t=T_2(N_2)+1}^{\infty} \mathbb{I}\left\{E_1(t)\right\}\right] \mathbb{I}\left\{T_2(N_2) < \infty\right\}.
\end{aligned}$$

Note that the argument in Step 4 implies that for $t > T_2(N_2)$ the set J_t of plausible parameters contains only the true parameter θ_0. For this realization the PEL algorithm follows an optimal policy π_{θ_0} from time $T_2(N_2)$ onward, therefore $\sum_{t=T_2(N_2)+1}^{\infty} \mathbb{I}\left\{E_1(t)\right\} = 0$. Hence,

$$N_1 \geq \sum_{t=0}^{\infty} \mathbb{I}\left\{E_1(t)\right\} = \sum_{t=0}^{\infty} \mathbb{I}\left\{V^{\mathcal{A}_t}(h_t) < V_{\theta_0}^*(s_t) - \epsilon\right\},$$

where equality holds since $\theta_0 \in J_t$ for realization in E_4^c (see 4). Hence, by definition of the PMC, $N_1 \geq PMC(\epsilon)$.

Step 6: The bound $N_1 \geq PMC(\epsilon)$ holds on $h_\infty \in E_4^c \cap E_5^c \cap E_6^c$. But, by the union bound, $\mathbb{P}^{\mathcal{A},s_0}\left\{E_4^c \cap E_5^c \cap E_6^c\right\} \geq 1 - \delta$. Substituting N_2 and T_{eff} yields the inequality (2) with probability of at least $(1 - \delta)$. ∎

4.7 Discussion

As may be seen from Equation (2), the dependence of PEL's PMC bound on $|\Theta|$ is essentially linear. The following example shows that without further assumptions on the model, this linear dependence can not be improved upon by *any* learning algorithm.

Example 2 Consider an array of N one-armed bandits (Robbins, 1952) $b_1, ..., b_N$, each with a payoff of 0 (loss) or 1 (gain). It is known that exactly one bandit b^* that has a high gain probability of $p_{\max}$, and all others have a lower gain probability of $p_{\min} < p_{\max}$. The agent may play any single bandit b_i at each time step.

Given that the index of the best bandit is initially unknown, our model set contains N different models, namely $|\Theta| = N$. Consider $PMC(\epsilon)$ with ϵ small enough so that a policy mistake occurs each time the agent chooses a suboptimal bandit. Obviously, a learning agent needs to converge to the (initially unknown) bandit b^*. It is evident that any learning algorithm may need to try out all N bandits in order to find the best one; thus, the (worst-case) PMC is at least linear in $|\Theta|$ (see Mannor and Tsitsiklis (2004) for a stronger result).

5. Decomposable Models

As indicated above, the dependence of PEL's PMC bound on $|\Theta|$ is essentially linear. Unfortunately, $|\Theta|$ may turn out to be prohibitively large, especially when the parameter vector is multidimensional. To be specific, consider a system which consists of N coupled subsystems, each parameterized independently by a parameter $\theta^i \in \Theta^i$, where $i = 1, 2, ..., N$. The complete system is thus described by a parameter vector $\theta = (\theta^1, ..., \theta^N)$ from the set $\Theta = \times_{i=1}^N \Theta^i$, so that the size $|\Theta| = \prod_{i=1}^N |\Theta^i|$ of the composite parameter set grows exponentially in N. This may be viewed of course as a particular manifestation of the well known "curse of dimensionality."

However, if the subsystems concerned are decomposable in an appropriate sense, we can obtain separate statistical information about each component θ^i of the parameter vector independently. It is our purpose in this section to utilize such decomposable model structure in order to obtain better performance bounds on the AMC of our algorithm. Furthermore, we introduce a natural modification of the PEL algorithm to these models - the D–PEL algorithm. This algorithm maintains separate plausible sets and cumulative likelihood functions for each component θ^i. This reduces both the memory complexity of parameter bookkeeping from $\prod_{i=1}^N |\Theta^i|$ to $\sum_{i=1}^N |\Theta^i|$ and the computational complexity of finding the most likely parameter at the elimination step.

This section proceeds as follows. We formalize the notion of a decomposable model in Section 5.1. In Section 5.2 we introduce a version of the PEL algorithm, the Decomposable PEL (D-PEL) algorithm, fitted to these models. We provide a mistake bound for the D-PEL algorithm that is linear in the sum of the sizes of parameter sets describing each subsystem. We conclude with the proof of the mistake bound on the D-PEL algorithm in Section 5.3.

5.1 Decomposable Models: Definition

We next formally introduce the notion of *decomposable models*. We consider a system M consisting of N subsystems $M_1, ..., M_N$. Denote by $\mathbb{S}^i$ the state space of subsystem M_i and let $\mathbb{S} = \times_{i=1}^N \mathbb{S}^i$ be the composite state space of the model. Thus, the state $s \in \mathbb{S}$ is a vector $s = (s^1, ..., s^N)$ of subsystem states. We assume that the state transition of each subsystem M_i from state s^i to $\bar{s}^{\,i}$ occurs in two stages. At the first stage each subsystem generates a coupling variable $y^i \in \mathbb{Y}^i$, where $\mathbb{Y}^i$ is a finite set. We assume that probability distribution α_i of y_i depends on the current action a and on the current state s^i of the system M_i, i.e., $y_i \sim \alpha_i(\cdot|s^i, a)$. At the second stage, the next state $\bar{s}^{\,i}$ of each subsystem M_i is generated via a probability distribution β_i that depends on the current state s^i, the current action a and the vector $y = (y^1, ..., y^N)$ of coupling variables, i.e., $\bar{s}^{\,i} \sim \beta_i(\cdot|s^i, a, y)$. The joint distribution of the next state $\bar{s}$ and the coupling vector y may then be expressed as follows:

$$\mathbb{P}\left\{\, \bar{s}, y \,|\, s, a\right\} = \prod_{i=1}^N \beta_i(\bar{s}^{\,i}|s^i, a, y) \prod_{j=1}^N \alpha_i(y^j|s^j, a).$$

We further assume that each subsystem M_i produces a reward $r_i \in \mathbb{R}^i$, where $\mathbb{R}^i$ is a finite set. The reward r_i is generated via a distribution $\eta_i(\cdot|s^i, a)$ that depends on the current state s^i and the current action a. The reward of the complete system is a sum of rewards produced by all subsystems, i.e., $r = \sum_{i=1}^N r_i$. See Figure 1 for a graphical description of the dependencies above. Similar to standard MDPs, we denote by $M = \langle \mathbb{S}, \mathbb{Y}, \mathbb{A}, \mathbb{R}, \{\eta_i\}_{i=1}^N, \{\alpha\}_{i=1}^N, \{\beta\}_{i=1}^N \rangle$ the described decomposable model.

We further assume that the model parametrization is decomposable in the following sense. Let each subsystem M_i be parameterized with a parameters $\theta^i \in \Theta^i$, where the parameter defines distributions $\alpha_i(y^i|s^i, a)$, $\beta_i(\bar{s}^{\,i}|s^i, a, y)$ and $\eta_i(r^i|s^i, a)$. Let $\Theta = \times_{i=1}^N \Theta^i$ denote the composite parameter set and let $\theta \equiv (\theta^1, , ..., \theta^N)$ be the composite parame-

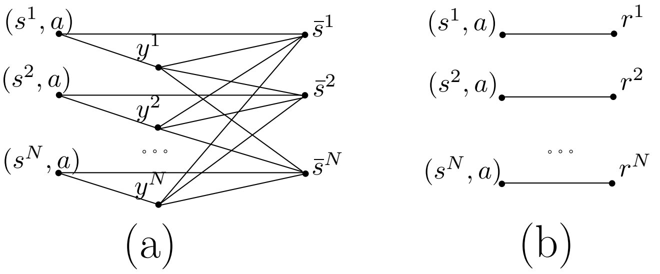

Figure 1: Graphical Representation of Decomposable Models.

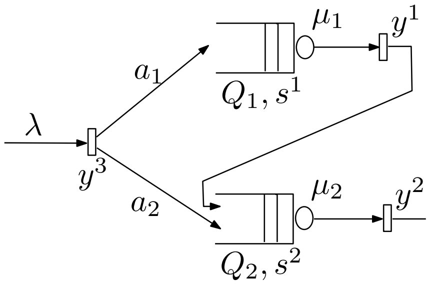

Figure 2: Two Concurrent Queues

ter vector. Denote $\alpha_i(y^i|s^i,a,\theta) = \alpha_i(y^i|s^i,a,\theta^i)$, $\beta_i(\bar{s}^{\,i}|s^i,a,y,\theta) = \beta_i(\bar{s}^{\,i}|s^i,a,y,\theta^i)$ and $\eta_i(r^i|s^i,a,\theta) = \eta_i(r^i|s^i,a,\theta^i)$.

We observe the following with respect to the proposed model:
1. The state dynamics of decomposable models is standard controlled Markov dynamics with the following state transition probabilities:

$$\mathbb{P}\{\bar{s}|\,s,a\} = \sum_{y\in\times_{i=1}^N \mathbb{Y}^i} \mathbb{P}\{\bar{s},y|\,s,a\} = \sum_{y\in\times_{i=1}^N \mathbb{Y}^i} \prod_{i=1}^N \beta_i(\bar{s}^{\,i}|s^i,a,y) \prod_{j=1}^N \alpha_i(y^j|s^j,a).$$

Further, the distribution of the immediate reward is independent of coupling vector y. Based on these observations, we conclude that there exists an optimal stationary policy $\pi : \mathbb{S} \to \mathbb{A}$ for this model that is independent of the value of coupling vector y. Hence, the optimal policy for decomposable models can be calculated using the same algorithms as for MDPs.
2. The augmentation of the basic state model by the (fully observed) coupling variables allows to decouple the parameter elimination in the different subsystems. The usefulness of this decomposition is illustrated by the following example:

Example 3 (Decoupling of Parameter Estimation) Consider two serially connected queues Q_1 and Q_2 equipped with servers S_1 and S_2 respectively. We assume that the service process of each server S_i is geometric with an unknown rate μ_i. The job arrival process is also assumed to be geometric with an unknown rate λ. The arrival and service processes are fully observed. New jobs arrive either to queue Q_1 or Q_2 depending on a control signal ($a = a_1$ or a_2, respectively). Jobs processed by the server S_2 exit the system, while jobs processed by the server S_1 are sent to the queue Q_2. See Figure 2 for schematic.

Denote by s^i be the number of jobs in queue Q_i, then the vector (s^1, s^2) fully defines the state of this system. We note that the standard one-stage state-transition dynamics fails to provide us with separate statistical information on each parameter. For example, for $s^1, s^2 > 0$ $P(\bar{s}^{\,1} = s^1, \bar{s}^{\,2} = s^2 | s^1, s^2, a_1) = (1-\lambda)(1-\mu_1)(1-\mu_2) + \lambda\mu_1\mu_2$, as this transition can occur in two different ways: (1) there were no arrival nor service in the last time step; (2)a new job arrived and an existing jobs was serviced. Hence, observing only the state vector (s^1, s^2) does not provide separate statistical information on each parameter in this case.

However, the state-transition dynamic of this system may be naturally divided into the following two stages. (1) Servers S_1, S_2 generate coupling variables y^1, y^2 indicating whether they finished processing a job in the current time step. The arrival process, in turn, generates a coupling variable y^3 indicating an arrival of a new task; (2) Given the coupling vector $y = (y^1, y^2, y^3)$ and the current number of jobs s^1 and s^2 in queues Q_1 and Q_2 respectively, the number of jobs $\bar{s}^{\,1}$ and $\bar{s}^{\,2}$ in queues in the following step is deterministic.

Given the current state of the system, the probability distribution of y^1 and y^2 depends only on μ_i and the probability distribution of y^3 depends only on λ:

$$\begin{aligned} p(\bar{s}^{\,1}, \bar{s}^{\,2}, y^1, y^2, y^3 | s^1, s^2, \lambda, \mu_1, \mu_2) &= p(y^3|\lambda) p(y^1|\mu_1, s^1) p(y^2|\mu_2, s^2) \\ &\quad \cdot p(\bar{s}^{\,1}|s^1, a, y^1, y^3) p(\bar{s}^{\,2}|s^2, a, y^1, y^2, y^3). \end{aligned}$$

Hence, observing both the state and the coupling vectors we are able to obtain statistical data on each parameter separately.

This subsection is concluded with an example of two well-known models that have a decomposable structure: factored models and Markov bandits. We begin by noting that decomposable models can be seen as a refinement of *factored models* introduced by Kearns and Koller (1999) and defined in the following way:

Example 4 (Factored Models) Let $M =< \mathbb{S}, \mathbb{A}, \mathbb{R}, p, \eta >$ be an MDP. Let the state space $\mathbb{S}$ be a product $\mathbb{S} = \times_{i=1}^{N} \mathbb{S}^i$, i.e., a state $s \in \mathbb{S}$ is a vector $s = (s^1, ..., s^N)$. We assume that for each $i \in \{1, ..., N\}$ and action $a \in \mathbb{A}$ there exists a subset $\mathrm{Pa}_a^i(s)$ of components of state s ("parents" of $\bar{s}^i$) so that

$$p(\bar{s}|s, a) = \prod_{i=1}^{N} p_i \left(\bar{s}^{\,i} | \mathrm{Pa}_a^i(s), a \right).$$

Namely, each component of the state $\bar{s}$ is drawn independently from the distribution that depends on the action a and a subset of components of the state s. We further assume that the reward function is a deterministic function that depends on the current state only. An MDP that admits such decomposition is called *factored.*

We consider a version of factored model parameterized with a set $\Theta = \times_{i=1}^{N} \Theta^j$. Namely, we assume that for each $i \in \{1, ..., N\}$ and each $a \in \mathbb{A}$ the distribution $P_i \left(\bar{s}^{\,i} | \mathrm{Pa}_a^i(s), a \right)$ is parameterized with single component θ^i of the parameter vector. This model can be seen as a decomposable model with coupling variables $y^i = \mathrm{Pa}_a^i(s)$. Here the coupling variables help to capture the dependence of $\bar{s}^{\,i}$ on several components of s rather than just on s^i.

The major advantage of factored models, is that the transition probabilities for each component of the state vector can be learned separately, thus reducing significantly the PMC of "flat" model-based algorithms (see Kearns and Koller (1999)). Computing the optimal policy can also be done more efficiently for factored model (Guestrin, 2003).

However, the component independence property of factored models may not hold even in a relatively simple models. For instance, consider the serial queueing system defined in

Example 3. On the one hand, as the state transitions in the queue are interdependent, the queue–size state vector (s^1, s^2) can not be factored in a straightforward manner as $p\left(\bar{s}|s,a\right) = p_1\left(\bar{s}^{\,1}|s^1,a\right)p_2\left(\bar{s}^{\,2}|s^2,a\right)$. On the other hand, we have shown that it can be represented as a decomposable model by adding the coupling variables.

We note, that similarly to factored model, the probability distribution α_i of each coupling variable y^i and the probability distribution β_i of each s^i in decomposable models can also be learned separately. We believe that a proper modification of "flat" model-based algorithms may yield a boost to the learning rate in decomposable models even in absence of parameterized model.

Example 5 (Markov Bandits) Consider the Markovian multi–armed bandit problem (Gittins, 1989). This model consists of N Markov processes $b_1, ..., b_N$, where the i-th process is a Markov chain C_i over a finite state space $\mathbb{S}^i$. If bandit b_i is at some state $s^i \in \mathbb{S}^i$ and is played, then a random reward r is received with probability $\eta_i(r|s^i)$, the state of this bandit changes to $\bar{s}^{\,i} \in \mathbb{S}^i$ with probability $p_i(\bar{s}^{\,i}|s^i)$, while the states of all the other bandits remain unchanged. The goal of the agent is to maximize the expected discounted reward by playing interchangeably the given N bandits.

Consider learning version of the Markov bandit problem with independent parametrization of bandits. Namely, the transition and reward probabilities of each bandit b_i depend on a parameter $\theta^i \in \Theta^i$. Hence the parameter set is $\Theta = \times_{i=1}^{N}\Theta^i$ and parameter values are vectors $\theta = (\theta^1, ..., \theta^N)$.

This model can be seen as a decomposable MDP with each bandit b_i as a sub-system with state space $\mathbb{S}^i$ and an empty set $\mathbb{Y}^i$ of values of coupling variable. It can be easily seen that both reward distribution and state transition probability function decompose as required.

5.2 The D-PEL Algorithm

The Decomposable PEL (D-PEL) algorithm proceeds as follows (see Algorithm 2 for details). Essentially, it performs parameter elimination for each component of the parameter vector separately. Namely, the algorithm maintains lists $J^i \subseteq \Theta^i$ of plausible values for each component θ^i of the parameter vector, where $i = 1, ..., N$. Initially, all parameter values are considered plausible (i.e., $J^i = \Theta^i$) and then they are eliminated sequentially. For brevity of notation we denote by $\mathbb{J} = \times_{i=1}^{N} J^i \subseteq \Theta$ the composite plausible parameters list.

Similarly to the PEL algorithm, the elimination step is based on the Sequential Probability Ratio Test (SPRT) with the following three modifications. First, we perform hypothesis testing of parameter values in each plausible list J^i separately. Second, we use a smaller value of the elimination threshold:

$$G_{th} = \log\left[\frac{3\sum_{i=1}^{N}(\left|\Theta^i\right| - 1)}{\delta}\right].$$

This reduction is due to smaller number of eliminations needed by the D-PEL algorithm until the identification of the true parameter. The maximal number of eliminations in the D-PEL algorithm is $\sum_{i=1}^{N}\left|\Theta^i\right| - N$ as opposed to $\prod_{i=1}^{N}\left|\Theta^i\right| - 1$ in the original PEL algorithm. Finally, we adjust the definition of the log-likelihood function of the observation $o_t \triangleq (s_{t-1}, a_{t-1}, r_{t-1}, y_t, s_t)$ to encapsulate the fact that we obtain statistical information on different components of the parameter vector separately.

We let $l_t^i(\theta^i)$ for $i \in \{1, ..., N\}$ and $\theta^i \in \Theta^i$ denote the log-likelihood of observation o_t with respect to the value θ^i of the component θ^i of the parameter vector. Namely,

$$l_t^i(\theta^i) = \log \alpha_i(y_t^i|s_{t-1}, a_{t-1}, \theta^i) + \log \beta_i(y_t^i|s_{t-1}, a_{t-1}, \theta^i) + \log \eta_i(r_t|s_{t-1}, a_{t-1}, \theta^i). \qquad (7)$$

The *cumulative* log-likelihood for the component θ^i is then $G_t^i(\theta^i, i) = \sum_{i=1}^{t} l_t^i(\theta^i)$.

The exploration-exploitation tradeoff is addressed in the same way as in the original PEL algorithm. At each time step t, the D-PEL algorithm selects an "optimistic" action in the following sense. First, the algorithm selects the parameter vector $\theta(t) \in \mathbb{J}_t$ that maximizes the value function $V_\theta^*(s_t)$ for the current state s_t. The selected action is then the optimal one given $\theta(t)$, i.e., $a_t = \pi_{\theta(t)}^*(s_t)$.

Algorithm 2 Decomposable Model Parameter ELimination

```
Input: $\{M_\theta\}_{\theta \in \Theta}$ – the finite family of possible decomposable models, $\delta$ – an allowed probability of error.
Initialize: Initialize the lists of plausible parameter values to $J_0^i = \Theta^i$ for all $i \in \{1, ..., N\}$. Initialize the arrays of cumulative log-likelihood to $G_0^i(\theta^i, i) = 0$ for all $i \in \{1, ..., N\}$ and $\theta^i \in \Theta^i$.
For $t = 0, 1, ...$ do
1. Stopping condition: If all lists $J_t^i$ are singletons, namely $J_t^i = \{\theta^i\}$, then use the corresponding policy $\pi_{(\theta^1, ..., \theta^N)}^*$ indefinitely and skip items (2)-(5) below.
2. Find an optimistic parameter: Select a parameter value that maximizes the value function among plausible parameter values: $\theta(t) = \arg\max_{\theta \in \mathbb{J}_t} V_\theta^*(s_t)$.
3. Act: Execute the action according to the optimal policy for the optimistic parameter: $a_t = \pi_{\theta(t)}^*(s_t)$.
4. Update: Observe the reward $r_t$, the vector $y_t$ of coupling variables and the next state $s_{t+1}$. Update for all $i \in \{1, ..., N\}$, $\theta^i \in J_t^i$: $G_{t+1}^i(\theta^i) = G_t^i(\theta^i) + l_{t+1}^i(\theta^i)$, where $l_{t+1}^i$ is defined in (7).
5. Eliminate: For all $i \in \{1, ..., N\}$ set $J_{t+1}^i = J_t^i$ and do:
    a. For all $\theta^i \in J_{t+1}^i$ so that $G_{t+1}^i(\theta^i) = -\infty$, let $J_{t+1}^i = J_{t+1}^i \setminus \{\theta^i\}$.
    b. Find the most likely parameter in the plausible set $\hat{\theta}^i = \arg\max_{\theta^i \in J_{t+1}^i} G_{t+1}^i(\theta^i)$.
    c. For all $\theta^i \in J_{t+1}^i$ so that
       $$G_{t+1}^i(\hat{\theta}^i) - G_{t+1}^i(\theta^i) > \log\left[\frac{3\sum_{i=1}^{N}(|\Theta^i| - 1)}{\delta}\right], \qquad (8)$$
       let $J_{t+1}^i = J_{t+1}^i \setminus \{\theta^i\}$.
```

The high probability bound on the mistake count of the D-PEL algorithm is given by the following theorem:

Theorem 19 *Consider the D-PEL algorithm with parameter* $0 < \epsilon < \frac{R_{\max}}{(1-\gamma)}$ *and* $0 < \delta < 1$. *Then with probability of at least* $1 - \delta$, *D-PEL's policy-mistake count is upper bounded by*

$$PMC(\epsilon) \le L(\epsilon, \delta, \gamma)\left[\sum_{i=1}^{N} |\Theta^i|\right] \frac{R_{\max}^3}{\epsilon^3(1-\gamma)^6} \log\left(\frac{3\sum_{i=1}^{N}|\Theta^i|}{\delta}\right), \qquad (9)$$

where $L(\epsilon, \delta, \gamma) = 1000 \log \frac{4R_{\max}}{\epsilon(1-\gamma)}$.

This theorem implies that the D-PEL algorithm is PPAC in terms of the total mistake bound, and its PMC is proportional (up to the logarithmic term) to the sum of sizes of the parameter sets Θ^i. When $\left|\Theta^1\right| = ... = \left|\Theta^N\right|$ the bound is linear in the number N of components of parameter vector as opposed to the original PEL algorithm whose PMC bound is exponential in N. Moreover, the amount of memory required for storage of lists of plausible parameters and cumulative likelihood functions is also reduced from exponential to linear in N since we do not need to keep track of log-likelihood and plausibility of all combinations $\theta = (\theta^1, ..., \theta^N)$ as in the original PEL algorithm.

Remark 20 Computational and memory complexities of calculating an optimistic parameter and optimistic policy remains exponential in N for a general model. However, these values can be computed in time linear in N in a variety of well-known models, including Markov Bandits, certain queuing systems and particular variants of the inventory problem.

Remark 21 The D–PEL algorithm can be applied to *any* parametrization that provides statistical information about each component of the parameter vector separately. For instance, in some models the parameter θ^i may be further decomposed to a triplet $(\theta^i_\alpha, \theta^i_\beta, \theta^i_\eta)$ so that $\alpha_i(y^i|s^i, a, \theta^i) = \alpha_i(y^i|s^i, a, \theta^i_\alpha)$, $\beta_i(\bar{s}^{\ i}|s^i, a, y, \theta^i) = \beta_i(\bar{s}^{\ i}|s^i, a, y, \theta^i_\beta)$ and $\eta_i(r^i|s^i, a, \theta^i) = \eta_i(r^i|s^i, a, \theta^i_\eta)$. D–PEL's analysis applies to these models and provides further gain in terms of a bound on D–PEL's PMC. The specific parametrization described in Section 5.1 was chosen for the purposes of the brevity of exposition.

5.3 Proof of Theorem 19

This section contains an outline of the proof of Theorem 19. The essential part of the proof is similar to the proof of Theorem 4, and we indicate explicitly only those parts that are substantially different.

We denote by $\theta_0 = (\theta^1_0, ..., \theta^N_0)$ the vector of the actual values of all parameters and consider the D-PEL algorithm applied to the true MDP M_{θ_0}. We note that similarly to the PEL algorithm, the elimination step of D-PEL can be interpreted as an SPRT test between θ^i and $\theta^{i'}$ with parameter

$$\delta' \triangleq \frac{\delta}{3\sum_{i=1}^{N}(|\Theta^i| - 1)}.$$

Denote by $R^i_t(\theta^i)$ the total Bhattacharyya information of the observed history h_t with respect to parameters θ^i and θ^i_0. Further denote by $N^B_i(\theta^i, \delta)$ and $N^W_i(\theta^i, \delta)$ the corresponding Bhattacharyya stopping time (see Definition 16) and the corresponding SPRT stopping time (see Definition 12).

We next modify Lemma 18 (the information count lemma) in the following way. Its proof can be found in Appendix F.

Lemma 22 *Consider a history* $h_\infty = \{s_t, a_t, r_t\}^\infty_{t=0}$ *so that the following two conditions are satisfied:*
(1) The true parameters θ^i_0*,* $i = 1...N$*, are not eliminated during the execution of the D-PEL algorithm;*
(2) $N^W_i(\theta^i, \delta') \leq N^B_i\left(\theta^i, (\delta')^{3/2}\right)$ *for all* $\theta^i \neq \theta^i_0$ *and all* $i = 1, ..., N$*, i.e., the relation between*

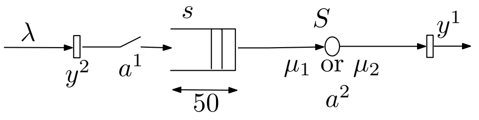

Figure 3: Simulated Model: Queue with admittance control and two-mode server.

Bhattacharyya stopping time and SPRT stopping time is as indicated by Lemma 17. Then the total number of visits to informative state-action pairs before all $\theta^i \neq \theta_0^i$, $i = 1, ..., N$ are eliminated is upper bounded by

$$\left(\sum_{i=1}^{N}(|\Theta^i| - 1)\right)\frac{\log((1/\delta')^{3/2})}{B_0},$$

where $B_0 \triangleq \frac{1}{8}\left(\frac{\epsilon(1-\gamma)^2}{4R_{\max}}\right)^2$.

6. Simulation Experiments

In this section we illustrate the theoretical convergence results for the PEL and D-PEL algorithms with some simulation experiments. As a reference point, we provide simulation results of a state-of-the-art model-based algorithm - the RTDP–IE algorithm (Strehl et al., 2006a). We show that by leveraging the parametrization our algorithms achieve a significantly smaller action mistake count.

In our experiments we consider a system composed of a discrete–time single queue with a buffer of size $B = 50$ equipped with a server S (see Figure 3). At each time step the dynamics of the system may be divided into the following two stages: (1) A new job arrives with probability λ. Depending on control signal a^1 it is either discarded ($a^1 = 1$) or admitted to the queue ($a^1 = 2$); If the queue buffer is full then the job is always discarded. (2) Depending on the control variable a^2 the job at the top of the queue may be processed by server S either in the slow mode ($a^2 = 1$) or in the fast mode ($a^2 = 2$). This job will be serviced and removed from the queue with probability μ_1 or μ_2 respectively, otherwise it will remain in the queue. The one-step reward is deterministic and is a sum of the following three terms: a cost of -1 for each job waiting in the queue, a cost of -2.7 for using mode 2 of the server and a fine of 9 for discarding a arriving job. We then normalize the reward so it would be in the interval $[0, 1]$ as it is required by the RTDP–IE algorithm. We further assume that $\mu_1, \mu_2, \lambda \in \mathcal{M}$, where $\mathcal{M} \triangleq \{0.1, 0.2, ..., 0.9\}$.

Let the number s number of jobs in the queue be the state of the system. We set the true parameters to be $\lambda = 0.3$, $\mu_1 = 0.5$ and $\mu_2 = 0.8$ and use a discount factor of $\gamma = 0.9$. We note that the optimal control policy for these values of parameters and values of costs described above is not trivial in the sense that it does not choose the same action for all states (see Figure 4).

We further denote by y^1 an observed binary variable that indicates whether the server S finished processing a job in the last step. Finally, denote by y^2 an observed binary variable that indicates an arrival of a new job. Then the joint probability of the next state s' and coupling variables decomposes as follows:

$$p(s', y^1, y^2|s, a^1, s^2, \lambda, \mu_1, \mu_2) = p(y^2|\lambda)p(y^1|s, \mu_i \text{ where } i = a^2)p(s'|s, a^1, y^1, y^2)$$

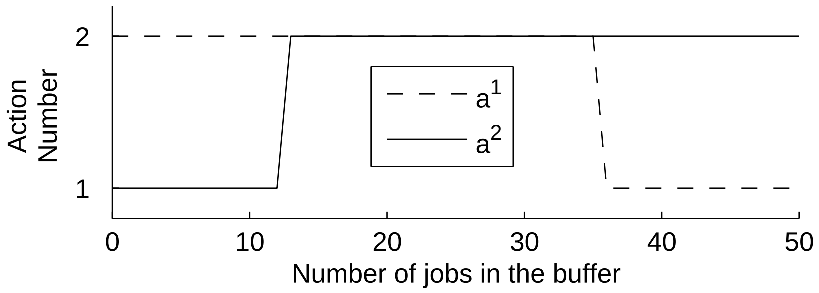

Figure 4: The optimal policy.

On the one hand, given variables (y^1, y^2), we can obtain statistical information on each of parameters μ_1, μ_2 and λ separately, hence the D-PEL algorithm and its analysis apply. On the other hand, this system may be modeled as a standard parameterized MDP, to which the PEL algorithm applies.

In order to make a fair comparison between PEL and D-PEL, we provide Step 4 ("Update") of the PEL algorithm with the same statistical information as the D-PEL has. Namely, we "allow" PEL to observe the coupling variables (y^1, y^2) for purposes of parameter elimination.

We stop all algorithms after 10 million steps and measure their performance by *empirical Action Mistake Count*, namely, the total number of suboptimal actions taken before stopping.

We tested the performance of the PEL and D–PEL algorithms for different values of the allowed probability of error δ, while simulation for each value was repeated 1000 times.

Figure 5 depicts the fraction of successful learning episodes for the PEL and the D-PEL algorithms, namely, simulations in which the true parameter was not eliminated. According to our analysis (see Step 1 in Section 4.6) this fraction is lower bounded by $1 - \delta/3$. However, the actual fraction is much larger. This observation can be explained by non-tightness of the union bound used in deriving this bound.

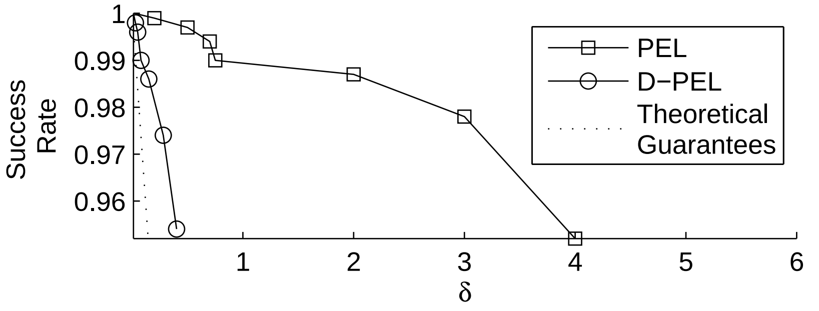

Figure 5: Fraction of successful learning episodes.

Figure 6 depicts the mean value of empirical AMC of the PEL and the D–PEL algorithms over successful learning episodes for different values of parameter δ. In order to make a fair comparison between performances of PEL and D–PEL, we plot the AMC of both algorithms as a function of their empirical probability δ^n of elimination of the true parameter. We refer to parameter δ^n as the *normalized* δ-parameter and it can be easily calculated as 1 minus the fraction of successful learning episodes given by Figure 5. For clarity of exposition we set the height of error bars to one third of empirical standard deviation.

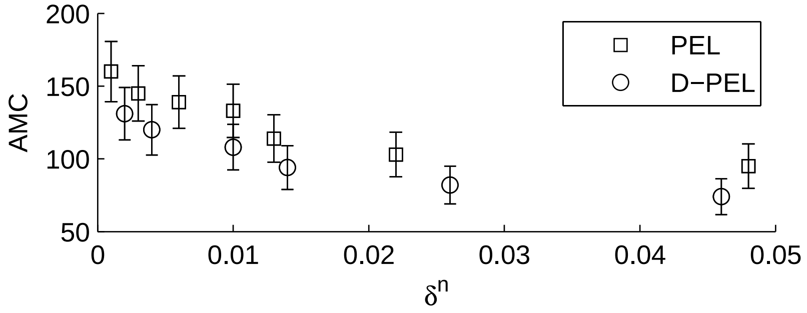

Figure 6: Empirical AMC of the PEL and the D–PEL algorithms.

Comparing the empirical AMC to the bounds given in Theorems 4 and 19, these bounds turn out to be significantly larger that the observed empirical AMC, namely, these bounds are loose. The main value of these bounds is in their functional form, namely, the form of the dependencies on different model and algorithm parameters.

For comparison, we apply the RTDP–IE algorithm to our model. This algorithm requires a user-defined parameter β that allows to tune the algorithm's solution to exploration–exploitation trade–off (see Section 4.2 in Strehl et al. (2006a)). We scanned a variety of values for this parameter and repeated simulation 1000 times for each value of β (see Figure 7). In order to make a fair comparison to the performances of the PEL and the D–PEL algorithms we calculate the mean empirical AMC for the RTDP–IE algorithm over the 700 repetitions that yielded the lowest AMC (out of total 1000 repetitions). For consistency with previous graphs the height of error bars is set to one third of empirical standard deviation. We observe that the minimal mean AMC is obtained for $\beta = 0.07$ and is equal to $1.06 \cdot 10^4$, which is about 50 times larger than the AMC of the PEL and the D-PEL algorithms.

7. Conclusion

Parameterized models offer a great potential for reduction of learning time and cost in large RL problems, alongside less structured methods such as function approximation, aggregation

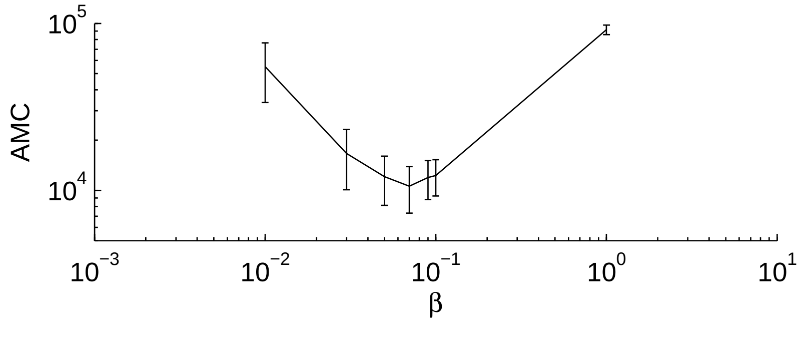

Figure 7: Empirical AMC of the RTDP-IE algorithm.

and state abstraction. The former should be used when the available prior information allows to reduce model uncertainty to a lower dimensional parameter space, thereby allowing explicit modeling of inter-state dependencies and avoiding the pitfalls inherent in the local nature of learning in the general unstructured model.

In this paper we have considered the case of parameterized models with with discrete parameters. We proposed a learning algorithm that incorporates efficient exploration to achieve polynomial mistake bounds in the PAC sense. As may be expected these bounds are independent of the cardinality of the state and action spaces, and in fact may well apply to continuous spaces under reasonable regularity conditions.

We further introduced a family of *decomposable* models which roughly correspond to a system consisting of several independently parameterized subsystems coupled through control and/or some exogenous coupling variables. We provided a variant of the PEL algorithm, the D-PEL algorithm, for efficient learning in these models. We further demonstrated that several well-known problems fit into decomposable model framework.

We concluded our paper with an experiment that shows that our algorithms learn faster than a state-of-the-art model based algorithm that does not exploit parametrization of the model.

Several choices were made in the design of proposed algorithms. First, the basic approach taken was that of parameter elimination, rather than on-line parameter estimation. Parameter elimination has the advantage of reducing the considered parameter set over time, which can quickly converge to a small set if sufficient statistical information is obtained. On the theoretical side, this approach allows the application of sequential hypotheses testing for the analysis of the algorithm. However, it should be realized that the possible error of eliminating the true parameter cannot be rectified later, and it is therefore important to keep its probability small. The second choice made in the algorithm is to incorporate an optimistic policy which is defined on a per-state basis, rather than freeze a stationary policy that is optimal for a certain parameter from a certain state. We believe this approach may add to exploration efficiency, although no direct comparison is available.

The main weakness of our algorithms is their computational and memory complexities. We need to calculate the optimal value function and the optimal policy for every possible parameter. As we have shown, in decomposable models the number of parameters may grow exponentially fast in the number of subsystems, hence so is the computational complexity for the general model. However, there are many standard models, including certain queuing systems and inventory problems, where the complexity of this computation grows only linearly in the number of subsystems. Furthermore, in some of these models the optimal value function and the optimal policy can be expressed analytically as a closed-form function, thus making the computational burden of finding optimal policies trivial.

The PEL and the D–PEL algorithms may be extended to the continuous parameter case through discretization. In Chapter 5 in Dyagilev (2009) we provide this extension for the PEL algorithm and analyze its performance. This algorithm is also elimination-based, however, it relies on a modified version of the sequential probability ratio test that takes into account the discretization error that might bias the likelihood ratios. The total mistake bound of the continuous parameter PEL algorithm is shown to be linear in the number of points on the grid needed for discretization. The D–PEL algorithm may be extended in a similar manner.

We further note that the analysis of both our algorithms relies on the fact that the actual model belongs the given set of models. It is possible to consider a weaker assumption that the state transition and reward probabilities of the true model are sufficiently close, but not necessarily equal, to one of these models. Using ideas similar to these described in the previous paragraph, we can extend the PEL and the D–PEL algorithms to this setting without a significant change in performance guarantees. We can also relax the assumption that the optimal value and the optimal policy can be calculated exactly. Our analysis may be easily modified to account for sub-optimality of the calculated policy and imprecision in the calculation of its value.

References

K.J. Astrom and B. Wittenmark. *Adaptive Control: Second Edition.* Dover Publications Inc., 1995.

A. Bernstein. Adaptive state aggregation for reinforcement learning. Master's thesis, Technion – Israel Institute of Technology. Available electronically via `http://tx.technion.ac.il/∼andreyb/MSc_Thesis_final.pdf`, 2007.

A. Bernstein and N. Shimkin. Adaptive aggregation for reinforcement learning with efficient exploration: Deterministic domains. In *Twenty-first Annual Conference on Learning Theory (COLT)*, pages 323–334. Omicron, July 2008.

D. P. Bertsekas and J.N. Tsitsiklis. *Neuro-dynamic Programming.* Athena Scientific, Belmont, MA, 1996.

R. I. Brafman and M. Tennenholtz. R-MAX - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231, 2002.

K. Dyagilev. Efficient reinforcement learning in parameterized models. Master's thesis, Technion – Israel Institute of Technology. Available electronically via `http://www.ee.technion.ac.il/people/shimkin/PREPRINTS/MScThesisByKirillDyagilev.pdf`, 2009.

K. Dyagilev, S. Mannor, and N. Shimkin. Efficient reinforcement learning in parameterized models: Discrete parameter case. In *Recent Advances in Reinforcement Learning: 8th European Workshop, Ewrl 2008, Villeneuve D'ascq, France, June 30-july 3, 2008, Revised and Selected Papers*, page 27. Springer, 2009.

J.C. Gittins. *Multi–Armed Bandit Allocation Indices.* John Wiley & Sons, Inc., New York, NY, USA, 1989.

C. Guestrin. *Planning Under Uncertainty in Complex Structured Environments.* PhD thesis, Stanford University, Stanford, CA, 2003.

T. Kailath. The divergence and Bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, 15(1):52–60, 1967.

S. M. Kakade. *On the Sample Complexity of Reinforcement Learning.* PhD thesis, Gatsby Computational Neuroscience Unit, University College London, UK, 2003.

M. Kearns and D. Koller. Efficient reinforcement learning in factored MDPs. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI 99)*, pages 740–747, 1999.

M. Kearns and S. P. Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49:209–232, 2002.

C.H. Kraft. Some conditions for consistency and uniform consistency of statistical procedures. *University of California Publications in Statistics*, 1:125–142, 1955.

P. R. Kumar and P. Varaiya. *Stochastic systems: estimation, identification and adaptive control*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1986.

S. Mannor and J.N. Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5:623–648, 2004.

W. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley-Interscience, Hoboken, NJ, 2007.

M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, 1994.

H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, pages 527–535, 1952.

A. L. Strehl and M. L. Littman. A theoretical analysis of model-based interval estimation. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*, pages 857–864, 2005.

A. L. Strehl, L. Li, and M. L. Littman. Incremental model-based learners with formal learning-time guarantees. In *Proceedings of the 22nd International Conference on Uncertainty in Artificial Intelligence (UAI 2006)*, pages 485–493, 2006a.

A. L. Strehl, E. Wiewiora, J. Langford, and M. L. Littman. PAC model-free reinforcement learning. In *Proceedings of the 23nd International Conference on Machine Learning (ICML 2006)*, pages 881–888, 2006b.

A.L. Strehl, L. Li, and M.L. Littman. Reinforcement learning in finite MDPs: PAC analysis. *Journal of Machine Learning Research*, 10:2413–2444, 2009.

R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, 1998.

A. Wald. *Sequential Analysis*. John Wiley & Sons, Inc., New York, NY, 1952.

Appendix A: Proof of Proposition 6

Let us consider the difference of the two value functions. Noting the definition of $\theta(J, s)$ and π_J, and substituting the corresponding Bellman backup we have

$$\begin{aligned} V_J(s) - V_\theta(s) &\geq V_J(s) - V_{\theta(J,s)}(s) \\ &= \gamma \sum_{s' \in S} p_J(s'|s, \pi_J(s)) \left[V_J(s') - V_{\theta(J,s)}(s') \right] \\ &\geq \gamma \sum_{s' \in S} p_J(s'|s, \pi_J(s)) \left[V_J(s') - V_{\theta(J,s')}(s') \right]. \end{aligned}$$

Repeating the argument n times we obtain that (with $s_0 \equiv s$),

$$\begin{aligned} V_J(s) - V_\theta(s) &\geq \gamma^n \sum_{s_1, s_2, \ldots, s_n \in S^n} \left(\prod_{i=1}^{n} p_J(s_i|s_{i-1}, \pi_J(s_{i-1})) \right) \cdot \left[V_J(s_n) - V_{\theta(J,s_n)}(s_n) \right] \\ &\geq -\frac{R_{\max}}{1-\gamma} \gamma^n \sum_{s_1, s_2, \ldots, s_n \in S^n} \left(\prod_{i=1}^{n} p_J(s_i|s_{i-1}, \pi_J(s_{i-1})) \right) \\ &\geq -\frac{R_{\max}}{1-\gamma} \gamma^n \xrightarrow{n \to \infty} 0. \end{aligned}$$

The second inequality follows since the value functions are positive and upper bounded by $R_{\max}/(1-\gamma)$. The third inequality uses the fact that sum of probabilities over all possible histories is equal to 1. ■

Appendix B: Proof of Proposition 10

Let $J = J_t$ denote the set of plausible parameters at time t and let π_J and $\theta(J, s)$ be defined as in Subsection 4.1. Denote by $K = K_t$ the set of non-informative state-action pairs at time t. Then

$$V_{\theta_0}^{\mathcal{A}}(h_t) \equiv \mathbb{E}^{\mathcal{A}, s_0} \left\{ \sum_{j=t}^{\infty} \gamma^{j-t} r_j \middle| h_t \right\} \geq \mathbb{E}^{\mathcal{A}, s_0} \left\{ \mathbb{I}\left\{E_3^c\right\} \sum_{j=t}^{t+T_{\text{eff}}-1} \gamma^{j-t} r_j \middle| h_t \right\},$$

where E_3^c denotes the event complementary to E_3. We wish to replace the policy $\mathcal{A}$ in the last expression with a stationary policy. For that purpose, define an auxiliary MDP M' which coincides with M_{θ_0} on $(s, a) \in K$ and with M_J (see Subsection 4.1) on $(s, a) \notin K$. Denote by $\mathbb{P}_{M'}^{\pi_J} \{\cdot | h_t\}$ the probability on sequence $(a_i, r_i, s_{i+1})_{i=t}^{\infty}$ induced by the policy π_J on M', with $\mathbb{E}_{M'}^{\pi_J} \{\cdot | h_t\}$ the corresponding expectation operator. The t-history h_t determines the sets J, G and K for this auxiliary process.

For any realization in $E_3^c(t)$, the set of plausible parameters J_τ is constant on the interval $\tau \in \{t, ..., t + T_{\text{eff}}\}$, hence the PEL algorithm follows the stationary policy π_J on that interval. Moreover, over that interval the PEL algorithm visits only state-action pair in K, hence the measure under MDP M_{θ_0} coincides with the measure under M' there. Therefore

$$\mathbb{E}^{\mathcal{A}, s_0} \left\{ \mathbb{I}\left\{E_3^c(t)\right\} \sum_{j=t}^{t+T_{\text{eff}}-1} \gamma^{j-t} r_j \middle| h_t \right\} = \mathbb{E}_{M'}^{\pi_J} \left\{ \mathbb{I}\left\{E_3^c(t)\right\} \sum_{j=t}^{t+T_{\text{eff}}-1} \gamma^{j-t} r_j \middle| h_t \right\}.$$

Substituting in the previous inequality we obtain:

$$\begin{aligned} V_{\theta_0}^{\mathcal{A}}(h_t) &\geq \mathbb{E}_{M'}^{\pi_J}\left\{ \mathbb{I}\left\{E_3^c(t)\right\} \sum_{j=t}^{t+T_{\text{eff}}-1} \gamma^{j-t} r_j \middle| h_t \right\} \\ &= \mathbb{E}_{M'}^{\pi_J}\left\{ \sum_{j=t}^{t+T_{\text{eff}}-1} \gamma^{j-t} r_j \middle| h_t \right\} - \mathbb{E}_{M'}^{\pi_J}\left\{ \mathbb{I}\left\{E_3(t)\right\} \sum_{j=t}^{t+T_{\text{eff}}-1} \gamma^{j-t} r_j \middle| h_t \right\}. \end{aligned}$$

The first term is a finite horizon value function $V_{M'}^{\pi_J}(s_t, T_{\text{eff}})$, while the sum in the second expectation can be bounded from above by $R_{\max}/(1-\gamma)$. Hence,

$$V_{\theta_0}^{\mathcal{A}}(h_t) \geq V_{M'}^{\pi_J}(s_t, T_{\text{eff}}) - \frac{R_{\max}}{1-\gamma} \mathbb{P}_{M'}^{\pi_J}\left\{ E_3(t) \middle| h_t \right\}. \tag{10}$$

Due to Lemma 6, 7 and 8, the first term satisfies

$$V_{M'}^{\pi_J}(s_t, T_{\text{eff}}) \geq V_{M'}^{\pi_J}(s_t) - \frac{\epsilon}{4} \geq V_J(s_t) - \frac{\epsilon}{2} \geq V_{\theta_0}(s_t) - \frac{\epsilon}{2}.$$

For the second term in (10), note that $\mathbb{P}_{M'}^{\pi_J}\left\{ E_3^c(t) \middle| h_t \right\} = \mathbb{P}^{\mathcal{A},s_0}\left\{ E_3^c(t) \middle| h_t \right\}$, hence $\mathbb{P}_{M'}^{\pi_J}\left\{ E_3(t) \middle| h_t \right\} = \mathbb{P}^{\mathcal{A},s_0}\left\{ E_3(t) \middle| h_t \right\}$. Thus,

$$V_{\theta_0}^{\mathcal{A}}(h_t) \geq V_{\theta_0}(s_t) - \frac{\epsilon}{2} - \frac{R_{\max}}{1-\gamma} \mathbb{P}^{\mathcal{A},s_0}\left\{ E_3(t) \middle| h_t \right\}.$$

On the other hand, for h_t in $E_1(t)$ the time step t is ϵ-suboptimal, namely $V_{\theta_0}^{\mathcal{A}}(h_t) < V_{\theta_0}(s_t) - \epsilon$. Combined with the previous inequality we obtain $\mathbb{P}^{\mathcal{A},s_0}\left\{ E_3(t) \middle| h_t \right\} > \frac{(1-\gamma)\epsilon}{2R_{\max}}$. ∎

Appendix C: Proof of Lemma 11

The proof of Lemma 11 is complicated by two facts. First, the events involved are not independent. Second, we need to consider only those time instances over which the probability to reach an informative state-action pair exceeds some threshold. Indeed, applying a concentration inequality (such an Hoeffding's or Azuma's) to all time instances, including those where this probability is null or very small, would result in a weak bound. The proposed solution is to apply an appropriate concentration inequality over an appropriate subsequence of (stopping) times.

This argument proceeds through the following proposition (see pp. 95–100 in Bernstein (2007) for the complete proof):

Proposition 23 (Abstract Discovery Lemma) *Denote by $\{\mathcal{F}_t\}$ a given filtration (i.e., an increasing sequence of σ-algebras) and by $\{D_t\}$ a sequence of events with $D_t \in \mathcal{F}_t$. Let*

$$Z \triangleq \sum_{t=1}^{\infty} \mathbb{I}\left\{ \mathbb{P}\left\{ D_t \middle| \mathcal{F}_{t-1} \right\} > p \right\},$$

where $p > 0$ is some given constant. Further, suppose that

$$\mathbb{P}\left\{ \sum_{t=1}^{\infty} \mathbb{I}\left\{D_t\right\} \leq M \right\} = 1$$

for some integer $M > 0$*. Then, for* $0 < \delta < 1$*,*

$$\mathbb{P}\left\{Z \leq \frac{2}{p}\left(M + \frac{4}{p}\log\frac{1}{\delta}\right)\right\} \geq 1 - \delta.$$

Let K_t be as in Definition 9 and let N_2 be a positive integer. Recall the definitions of $E_1(t)$, $E_2(t)$, $E_3(t)$ and $\mathcal{F}_t$ from Section 4.2.

Proof of Lemma 11: Define the following *discovery event* for $t \geq 0$:

$$D(t) \overset{\triangle}{=} \{\theta_0 \in J_t\} \bigcap \left\{\sum_{k=1}^{t} \mathbb{I}\{E_2(k)\} < N_2\right\} \bigcap E_3(t).$$

This event implies the following: by time step t the representative of the true parameter wasn't eliminated, and the informative event E_2 was encountered less than N_2 times; furthermore, in the following T_{eff} steps an least one additional event E_2 will occur. Note that $D(t) \in \mathcal{F}_{t+T_{\text{eff}}}$.

In order to employ Proposition 23 let us sample the series of events $D(t)$ and sigma-algebras $\mathcal{F}_t$ with the step of T_{eff}, i.e., for $i = 0, 1, 2, ...$ and $j \in \{0, .., (T_{\text{eff}} - 1)\}$ denote $D_{i+1}^{(j)} = D(i \cdot T_{\text{eff}} + j)$ and $\mathcal{F}_i^{(j)} = \mathcal{F}_{i \cdot T_{\text{eff}} + j}$. Note that $D_i^{(j)} \in \mathcal{F}_i^{(j)}$. For j as above define

$$Z^{(j)} \overset{\triangle}{=} \sum_{i=0}^{\infty} \mathbb{I}\left\{\mathbb{P}^{\mathcal{A},s_0}\left\{D_i^{(j)} \middle| \mathcal{F}_i^{(j)}\right\} > \frac{\epsilon(1-\gamma)}{2R_{\max}}\right\},$$

and note that

$$\mathbb{P}^{\mathcal{A},s_0}\left\{\sum_{i=1}^{\infty} \mathbb{I}\left\{D_i^{(j)}\right\} \leq N_2\right\} = 1$$

by the definition of $D_i^{(j)}$. Noting the definition of N_1, application of Proposition 23 with $p = \frac{\epsilon(1-\gamma)}{2R_{\max}}$ yields

$$\mathbb{P}^{\mathcal{A},s_0}\left\{Z^{(j)} \leq \frac{N_1}{T_{\text{eff}}}\right\} \geq 1 - \frac{\delta_3}{T_{\text{eff}}}.$$

Applying the union bound we obtain

$$\mathbb{P}^{\mathcal{A},s_0}\left\{\sum_{j=0}^{T_{\text{eff}}-1} Z^{(j)} \leq N_1\right\} \geq 1 - \delta_3.$$

We conclude the proof by showing that

$$\sum_{t=0}^{T_2(N_2)} \mathbb{I}\{E_1(t)\} \leq \sum_{j=0}^{T_{\text{eff}}-1} Z^{(j)} \equiv \sum_{t=0}^{T_2(N_2)} \mathbb{I}\left\{\mathbb{P}^{\mathcal{A},s_0}\{D_t | \mathcal{F}_t\} > \frac{\epsilon(1-\gamma)}{2R_{\max}}\right\}. \tag{11}$$

For some $t < T_2(N_2)$ let h_t be a t-history that satisfies $E_1(t)$ (if such history exists). For this history, $\theta_0 \in J_t$ by definition of $E_1(t)$ and the inequality

$$\sum_{k=1}^{t} \mathbb{I}\{E_2(k)\} < N_2$$

holds by definition of $T_2(N_2)$, hence the discovery event $D(t)$ occurs if and only if the event $E_3(t)$ occurs. Then, by Proposition 10,

$$\mathbb{P}^{\mathcal{A},s_0}\left\{ D_t | \, h_t \right\} > \frac{\epsilon(1-\gamma)}{2R_{\max}},$$

therefore

$$\mathbb{I}\left\{ P^{\mathcal{A},s_0}\{D_t|\mathcal{F}_t\} > \frac{\epsilon(1-\gamma)}{2R_{\max}} \right\} \geq \mathbb{I}\left\{E_1(t)\right\}$$

almost surely. Hence 11 is established and the claim follows.

Appendix D: Proof of Lemma 15

Kraft (1955) showed the following relation: $\frac{1}{2}\left\|p-q\right\|_1 \leq \sqrt{1-\rho^2}$. Equivalently, $\rho \leq \sqrt{1-\frac{1}{4}\left\|p-q\right\|_1^2}$, hence

$$\log\rho \leq \frac{1}{2}\log(1-\frac{1}{4}\left\|p-q\right\|_1^2) \leq -\frac{1}{8}\left\|p-q\right\|_1^2,$$

where the last inequality follows since $\log(1-x) \leq -x$ for all $0 \leq x < 1$. ∎

Appendix E: Proof of Proposition 17

Assume that H_0 holds true (the proof is identical under H_1). Let us consider histories x_0^n that lead to stopping of $N^B(\delta^{3/2})$. Since $p_1(x_n|x_0^{n-1}) = 0$ implies $N^W = N^B(\delta^{3/2})$ (if not stopped before), we can focus in the remainder of the proof only on stopping due to the first condition in (6). Let $N_1 = N^B(\delta^{3/2})$ and denote the log likelihood ratio of the history up to the stopping time N_1 as:

$$L\left(x_0^{N_1}\right) \triangleq \sum_{t=1}^{N_1} \log\frac{p_1(x_t|x_0^{t-1})}{p_0(x_t|x_0^{t-1})}.$$

Then $N^W(\delta) > N_1$ implies that $L\left(x_0^{N_1}\right) > \log\delta$, hence

$$\mathbb{P}\left\{N^W(\delta) > N_1\right\} \leq \mathbb{P}\left\{L\left(x_0^{N_1}\right) > \log\delta,\ N_1 < \infty\right\}.$$

Chernoff's inequality now implies

$$\mathbb{P}\left\{L\left(x_0^{N_1}\right) > \log\delta,\ N_1 < \infty\right\} \leq \mathbb{E}\left\{\exp\left\{\frac{1}{2}\left[L\left(x_0^{N_1}\right) - \log\delta\right]\right\}\mathbb{I}_{\{N_1<\infty\}}\right\},$$

hence

$$\mathbb{P}\left\{N^W(\delta) > N_1\right\} \leq \frac{1}{\sqrt{\delta}}E_C, \tag{12}$$

where

$$E_C \triangleq \mathbb{E}\left\{\exp\left\{\frac{1}{2}L\left(x_0^{N_1}\right)\right\}\mathbb{I}_{\{N_1<\infty\}}\right\}.$$

We proceed to bound E_C. Denote

$$d(x_{t+1}|x_0^t) \triangleq p_1^{1/2}(x_{t+1}|x_0^t)p_0^{1/2}(x_{t+1}|x_0^t)$$

so that $\rho(x_0^t) = \sum_{x' \in S} d(x'|x_0^t)$. Further denote $D(x_0^k) \triangleq \prod_{t=1}^k d(x_t|x_0^{t-1})$. Let Q_B to be the collection of N_1-histories $x_0^{N_1}$ for which $N_1 < \infty$, namely

$$Q_B = \left\{ x_0^k \in S^{k+1} \middle| \prod_{i=0}^{k-2} \rho(x_0^i) > \delta^{3/2} \text{ and } \prod_{i=0}^{k-1} \rho(x_0^i) \le \delta^{3/2} \right\}.$$

Substituting the definition of the expected value we obtain:

$$\begin{aligned} E_C &= \sum_{x_0^{N_1} \in Q_B} \exp\left\{ \frac{1}{2} \sum_{t=1}^{N_1} \log \frac{p_1(x_t|x_0^{t-1})}{p_0(x_t|x_0^{t-1})} \right\} \prod_{t=1}^{N_1} p_0(x_t|x_0^{t-1}) \\ &= \sum_{x_0^{N_1} \in Q_B} \prod_{t=1}^{N_1} \left(\frac{p_1(x_t|x_0^{t-1})}{p_0(x_t|x_0^{t-1})} \right)^{1/2} p_0(x_t|x_0^{t-1}) \\ &= \sum_{x_0^{N_1} \in Q_B} \prod_{t=1}^{N_1} p_1^{1/2}(x_t|x_0^{t-1}) p_0^{1/2}(x_t|x_0^{t-1}) \\ &= \sum_{x_0^{N_1} \in Q_B} D(x_0^{N_1}). \end{aligned}$$

In Dyagilev (2009) pp. 40–42 we show that

$$E_C \equiv \sum_{x_0^{N_1} \in Q_B} D(x_0^{N_1}) \le \sup_{x_0^{N_1} \in Q_B} \left\{ \prod_{t=0}^{N_1-1} \rho(x_0^t) \right\} \le \delta^{3/2}, \tag{13}$$

where the last inequality holds by definition of N_1. Thus, from (12) and (13),

$$\mathbb{P}\left\{ N^W(\delta) > N_1 \right\} \le \frac{1}{\sqrt{\delta}} E_C \le \delta.$$

■

Appendix F: Proof of Lemma 22

Recall the notations introduced in Section 5.3. Similarly to Lemma 18, we note that the elimination time of any $\theta^i \neq \theta_0^i$ for any i can be bounded from above by $N_i^W(\theta^i, \delta')$ which is dominated by $N_i^B\left(\theta^i, (\delta')^{3/2}\right)$ by condition (2). In what follows we bound the total number of visits to informative state-action pairs before the time $\max_{i=1,\dots,N} \max_{\theta^i \neq \theta_0^i} N_i^B\left(\theta^i, (\delta')^{3/2}\right)$ stops, thus bounding the number of these visits until all $\theta^i \neq \theta_0^i$ for all $i = 1, .., N$ are eliminated from the plausible sets J^i.

Let t be a time step on which an informative state-action pair (s_t, a_t) (see Definition 9) is visited. We let $\theta^i(t)$ denote the i-th component of the optimistic parameter vector at time t (see Algorithm 2). Let us assess the Bhattacharyya distance $(-\log \rho_t)$ between the joint distribution of (r_t, y_t, s_{t+1}) under the true model M_{θ_0} and the auxiliary model M_J. Evidently, it equals the sum of Bhattacharyya distances between $\eta_i(\cdot|s_t, a_t, \theta^i(t))$ and $\eta_i(\cdot|s_t, a_t, \theta_0^i)$, between $\alpha_i(\cdot|s_t, a_t, \theta^i(t))$ and $\alpha_i(\cdot|s_t, a_t, \theta_0^i)$ and between $\beta_i(\cdot|s_t, y_t, a_t, \theta^i(t))$ and $\beta_i(\cdot|s_t, y_t, a_t, \theta_0^i)$. Namely,

$$-\log \rho_t \quad = \quad \sum_{i=1}^{N} \left(-\log \rho_t^i \right),$$

where

$$\begin{aligned} -\log \rho_t^i \quad \triangleq \quad & -\log \left[\sum_{y^i \in \mathbb{Y}^i} \left(\alpha_i(y^i|s_t, a_t, \theta^i(t)) \alpha_i(y^i|s_t, a_t, \theta_0^i) \right)^{1/2} \right] \\ & -\log \left[\sum_{r^i \in \mathbb{R}^i} \left(\eta_i(r^i|s_t, a_t, \theta^1(t)) \eta_i(r^i|s_t, a_t, \theta_0^1) \right)^{1/2} \right] \\ & -\log \left[\sum_{s'^i \in \mathbb{S}^i} \left(\beta_i(s'^i|s_t, y_t, a_t, \theta^i(t)) \beta_i(s'^i|s_t, y_t, a_t, \theta_0^i) \right)^{1/2} \right]. \end{aligned}$$

Since $(s_t, a_t) \notin K_t$, then as in proof of Lemma 18 we obtain

$$-\log \rho_t > \frac{1}{8} \left(\frac{\epsilon(1-\gamma)^2}{4R_{\max}} \right)^2 \triangleq B_0.$$

We further note that $R_{t+1}^i(\theta^i(t)) - R_t^i(\theta^i(t)) = -\log \rho_t^i$ for $i = 1, ..., N$. Hence, by the pigeon-hole principle, the number of visits to an informative state-action pair until the times $N_i^B \left(\theta^i, (\delta')^{3/2} \right)$ stop is upper bounded by

$$\sum_{i=1}^{N} (\left|\Theta^i\right| - 1) \log((1/\delta')^{3/2}) / B_0.$$

■