Proceedings of the 2007 IEEE Symposium on Approximate
Dynamic Programming and Reinforcement Learning (ADPRL 2007)

Model-Based Reinforcement Learning in Factored-State MDPs

Alexander L. Strehl
Department of Computer Science
Rutgers University
110 Frelinghuysen Road
Piscataway, NJ 08854-8019
Email: strehl@cs.rutgers.edu

***Abstract*— We consider the problem of learning in a factored-state Markov Decision Process that is structured to allow a compact representation. We show that the well-known algorithm, factored Rmax, performs near-optimally on all but a number of timesteps that is polynomial in the size of the compact representation, which is often exponentially smaller than the number of states. This is equivalent to the result obtained by Kearns and Koller for their DBN-E^3 algorithm, except that we've conducted the analysis in a more general setting. We also extend the results to a new algorithm, factored IE, that uses the Interval Estimation approach to exploration and can be expected to outperform factored Rmax on most domains.**

I. Introduction

The development and analysis of the E^3 algorithm demonstrated that a *reinforcement-learning* (RL) agent can quickly (in polynomial time) obtain near-optimal behavior in an unknown, discrete-time Markov Decision Process (MDP) with finitely many states and actions [1]. Since then, there has been significant progress, including the introduction of the Rmax algorithm [2]. The Rmax and E^3 algorithms behave similarly, although Rmax is slightly simpler. The important work of [3] consolidated the theoretical framework and developed lower bounds. In a parallel movement, an advanced exploration technique called *Interval Estimation* (IE) that makes better use of the agent's experience was utilized in learning algorithms [4], [5]. It was shown that the Model-Based Interval Estimation algorithm learns in polynomial time and is superior (theoretically and empirically) to existing algorithms [6]–[8].

Unfortunately, in the worst case, any learning algorithm must visit every state and try every action before obtaining near-optimal return. For a given problem or application, it is often the case that easily obtained domain knowledge is available that restricts the class of MDPs in which the agent is to act. One convenient way to express this mathematically is with factored-state Markov Decision Processes, which allow different features of the state space to be modeled independently [9], [10]. Within this model, an algorithm similar to E^3 can learn after only examining a small fraction of the state space [11]. In fact, the algorithm learns after taking a number of actions that is only polynomial in the number of independent parameters of the model[1].

In this paper, we extend previous work in several ways. Like E^3, the Rmax algorithm has also been generalized to factored-state MDPs, but until now there has been no analysis of its performance [12]. Our first contribution is to provide such an analysis and prove formal bounds on its learning complexity. The extension of the E^3 and Rmax algorithms to Factored MDPs assumed that the transition dynamics of the system can be described by several *dynamic Bayesian networks*. We generalize this assumption significantly and conduct analysis in the more general framework. The work of [11] was primarily concerned with demonstrating that polynomial bounds are possible. We strive to keep the bounds as nearly tight as possible. In this respect, one important tool is the use of L1 bounds of the deviation of the empirical transition estimates to the true transition function. Finally, we also develop and analyze a new algorithm, called *Factored Interval Estimation*, for learning in Factored MDPs that utilizes the IE approach to exploration.

A. Related Work

Planning in a Factored MDP can be achieved by solving a linear program with a large number of variables and constraints. This formulation was examined in the work of [12], which used a linear approximation scheme to reduce the number of variables. In addition, a clever learning algorithm was developed that utilizes an approach similar to IE and explicitly deals with the problem of planning in the model.

II. Background and Notation

This section introduces the Markov Decision Process (MDP) notation used through out the paper; see [13] for an introduction. Let $\mathcal{P}_S$ denote the set of all probability distributions over the set S.

A **finite MDP** M is a five tuple $\langle \mathcal{S}, \mathsf{A}, T, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ is a finite set called the state space, A is a finite set called the action space, $T : \mathcal{S} \times \mathsf{A} \rightarrow \mathcal{P}_{\mathcal{S}}$ is the transition function, $\mathcal{R} : \mathcal{S} \times \mathsf{A} \rightarrow \mathcal{P}_{\mathbb{R}}$ is the reward function, and $0 \leq \gamma < 1$ is a discount factor on the summed sequence of rewards. We call the elements of $\mathcal{S}$ and A states and actions, respectively, and define $S = |\mathcal{S}|$ and $A = |\mathsf{A}|$. We use $T(s'|s,a)$ to denote the transition probability of state s' in the distribution $T(s,a)$

[1] The algorithm may use an exponential amount of computation, which is unavoidable in the worst case.

1-4244-0706-0/07/$20.00 ©2007 IEEE

and $R(s,a)$ to denote the expected value of the distribution $\mathcal{R}(s,a)$.

We assume that the learner (also called the *agent*) receives S, A, and γ as input. The *learning problem* is defined as follows. The agent always occupies a single state s of the MDP M. The agent is told this state and must choose an action a. It then receives an *immediate reward* $r \sim \mathcal{R}(s,a)$ and is transported to a *next state* $s' \sim T(s,a)$. This procedure then repeats forever. The first state occupied by the agent may be chosen arbitrarily. We define a *timestep* to be a single interaction with the environment, as described above.

A *policy* is any strategy for choosing actions. We assume (unless noted otherwise) that rewards all lie in the interval $[0,1]$. For any policy π, let $V^{\pi}_{M}(s)$ ($Q^{\pi}_{M}(s,a)$) denote the discounted, infinite-horizon value (action-value) function for π in M (which may be omitted from the notation) from state s. Specifically, for any state s and policy π, let r_t denote the tth reward received after following π in M starting from state s. Then, $V^{\pi}_{M}(s) = E[\sum_{t=0}^{\infty} \gamma^t r_t | s, \pi]$. This expectation is taken over all possible infinite paths the agent might follow. The optimal policy is denoted π^* and has value functions $V^*_M(s)$ and $Q^*_M(s,a)$. Note that a policy cannot have a value greater than $1/(1-\gamma)$.

A **factored-state MDP** (or **f-MDP**) is an MDP where the states are represented as vectors of n components $X = \{X_1, X_2, \ldots, X_n\}$. Each component X_i (called a **state variable** or **state factor**) may be one of finitely many values from the set $\mathcal{D}(X_i)$. In other words, each state can be written in the form $x = \langle x_1, \ldots, x_n \rangle$, where $x_i \in \mathcal{D}(X_i)$. The definition of factored-state MDPs is motivated by the desire to achieve learning in very large state spaces. The number of states of a factored-state MDP M is *exponential* in the number n of state variables. To simplify the presentation, *we assume the reward function is known and does not need to be learned.* All results can be extended to the case of an unknown reward function.

A. Restrictions on the Transition Model

Factored-state MDPs are most useful when there are restrictions on the allowable transition functions. Traditionally, researchers have studied transition models that can be represented by *dynamic Bayesian networks (DBNs)* for each action of the MDP. Such a representation has been shown to be powerful enough to support fast learning [11]. However, this representation is unable to model some important conditional independencies. Therefore, we develop a more general and more powerful model. For any factored state x, let x_i denote the ith component of x for $i = 1, \ldots, n$. Next, we introduce a model that yields a compact transition representation by allowing the transition probabilities for the factors of the next state x', $P(x'_i|x,a)$, to depend on only a subset of the state factors of the current state x.

Assumption 1: Let x, x' be two states of a factored-state MDP M, and let a be an action. The transition distribution function satisfies the following conditional independence condition:

$$T(x'|x,a) = \prod_i P(x'_i|x,a). \tag{1}$$

This assumption ensures that the values of each state variable after a transition are determined independently of each other, conditioned on the previous state. We consider transition functions that are structured as follows.

Definition 1: Let $\mathcal{I}$ be a set of **dependency identifiers**.

Definition 2: Let $D : \mathcal{S} \times \mathcal{A} \times X \rightarrow \mathcal{I}$ be a **dependency function**.

Assumption 2: Let $s, s' \in \mathcal{S}$ be two states and $a \in \mathcal{A}$ be an action. We assume that $P(s'_i|s,a) = P(s'_i|D(s,a,X_i))$. Thus, the transition probability from (s,a) to s' can be written as

$$T(s'|s,a) = \prod_{i=1}^{n} P(s'_i|D(s,a,X_i)) \tag{2}$$

The dependency function approach yields a compact representation of the underlying transition function by allowing commonalities among component distributions with shared dependency function behavior. It also generalizes other approaches, such as those using dynamic Bayes networks [11], and those incorporating decision trees [14] to represent abstraction. Several important definitions follow.

Definition 3: For each $(s,a) \in \mathcal{S} \times \mathcal{A}$, let

$$D_{s,a} := \{(X_i, j) \in X \times \mathcal{I} \mid j = D(s,a,X_i)\} \tag{3}$$

be the **relevant dependency pairs for state-action pair** (s,a).

Definition 4: Let $\mathcal{Q} = \cup_{(s,a) \in \mathcal{S} \times \mathcal{A}} D_{s,a}$ be the set of all **transition components**. Let N denote $|\mathcal{Q}|$, the number of transition components.

Note that each transition component, (X_i, j), corresponds to an independent probability distribution over the set $\mathcal{D}(X_i)$ that potentially needs to be estimated by the agent. We will provide algorithms whose learning complexity (as defined in Section II-B) depends only linearly on $\sum_{(X_i,j) \in \mathcal{Q}} |\mathcal{D}(X_i)|$, the number of parameters of the compact representation.

Definition 5: A **stochastic matrix** $X = \langle x(i,j) \rangle$ is an $m \times n$ matrix whose columns are probability vectors.

Definition 6: Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ be two n-dimensional vectors. The **L1 distance** between x and y, denoted $||x-y||_1$, is

$$||x-y||_1 := \sum_{i=1}^{n} |x_i - y_i| \tag{4}$$

B. Learning Efficiently

To formalize the notion of "efficient learning" we allow the learning algorithm to receive two additional inputs, ϵ and δ, both positive real numbers. The first parameter, ϵ, controls the quality of behavior we require of the algorithm (how close to optimality do we desire) and the second parameter, δ, is a measure of confidence (how certain do we want to be of the algorithm's performance). As these parameters decrease, greater exploration and learning is necessary, as more is expected of the algorithms.

An algorithm is simply a policy that, on each timestep, takes as input an entire history or trajectory through the MDP and outputs an action. To avoid explicitly mentioning histories in the following definition, we consider the set of policies (one on each timestep) executed by an algorithm.

Definition 7: (from [3]) For any fixed $\epsilon > 0$, the **sample complexity of exploration** (**sample complexity**, for short) of an algorithm $\mathcal{A}$ is the number of timesteps t such that the policy of the algorithm at time t, $\mathcal{A}_t$, is not ϵ-optimal from the current state, s_t at time t (formally, $V^{\mathcal{A}_t}(s_t) < V^*(s_t) - \epsilon$).

Next, we define what it means to be an "efficient" learning algorithm

Definition 8: An algorithm $\mathcal{A}$ is said to be an **efficient PAC-MDP** (Probably Approximately Correct in Markov Decision Processes) algorithm if, for any $\epsilon > 0$ and $0 \le \delta < 1$, the per-step computational complexity and the sample complexity of $\mathcal{A}$ are less than some polynomial in the relevant quantities $(S, A, 1/\epsilon, 1/\delta, 1/(1-\gamma))$, with probability at least $1 - \delta$.

Definition 9: An algorithm $\mathcal{A}$ is said to be a **PAC-fMDP** (Probably Approximately Correct in Factored Markov Decision Processes) algorithm if, for any $\epsilon > 0$ and $0 \le \delta < 1$, the sample complexity of $\mathcal{A}$ is less than some polynomial in the relevant quantities $(N, n, \max_i |\mathcal{D}(X_i)|, 1/\epsilon, 1/\delta, 1/(1-\gamma))$, with probability at least $1 - \delta$.

The terminology, PAC, in this definition is borrowed from [15], a classic paper dealing with supervised learning. Please see [3] for a full motivation of this performance measure. Note that in Definition 9, the sample complexity is allowed a dependence only on the number of parameters of the compact factored representation, which is often exponentially smaller than the number of states and actions

The following definitions and theorem were introduced and discussed in [16] and will be used in our analysis.

Definition 10: Suppose an RL algorithm $\mathcal{A}$ maintains an *action value*, denoted $Q(s,a)$, for each state-action pair (s,a) with $s \in S$ and $a \in \mathsf{A}$. Let $Q_t(s,a)$ denote the estimate for (s,a) immediately before the tth action of the agent. We say that $\mathcal{A}$ is a **greedy algorithm** if the tth action of $\mathcal{A}$, a_t, is $a_t := \text{argmax}_{a \in \mathsf{A}} Q_t(s_t, a)$, where s_t is the tth state reached by the agent.

Definition 11: Let $M = \langle \mathcal{S}, \mathsf{A}, T, R, \gamma \rangle$ be an MDP with a given set of action values, $Q(s,a)$ for each state-action pair (s,a), and a set K of state-action pairs. We define the **known state-action MDP** $M_K = \langle \mathcal{S} \cup \{S_{s,a} | (s,a) \notin K\}, \mathsf{A}, T_K, R_K, \gamma \rangle$ as follows. For each unknown state-action pair, $(s,a) \notin K$, we add a new state $S_{s,a}$ to M_K, which has self-loops for each action ($T_K(S_{s,a}|S_{s,a}, \cdot) = 1$). For all $(s,a) \in K$, $R_K(s,a) = R(s,a)$ and $T_K(\cdot|s,a) = T(\cdot|s,a)$. For all $(s,a) \notin K$, $R_K(s,a) = Q(s,a)(1-\gamma)$ and $T_K(S_{s,a}|s,a) = 1$. For the new states, the reward is $R_K(S_{s,a}, \cdot) = Q(s,a)(1-\gamma)$.

Definition 12: Suppose that for algorithm $\mathcal{A}$ there is a set of state-action pairs K_t (we drop the subscript t if t is clear from context) defined during each timestep t and that depends only on the history of the agent up to timestep t (before the (t)th action). Let A_K be the event, called the **escape event**, that some state-action pair (s,a) is experienced by the agent at time t, such that $(s,a) \notin K_t$.

Theorem 1: (from [16]) Let $\mathcal{A}(\epsilon, \delta)$ be any greedy learning algorithm such that for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let M_{K_t} be the known state-action MDP and π_t be the current greedy policy, that is, for all states s, $\pi_t(s) = \text{argmax}_a Q_t(s,a)$. Suppose that for any inputs ϵ and δ, with probability at least $1 - \delta$, the following conditions hold for all states s, actions a, and timesteps t: (1) $V_t(s) \ge V^*(s) - \epsilon$ (optimism), (2) $V_t(s) - V^{\pi_t}_{M_{K_t}}(s) \le \epsilon$ (accuracy), and (3) the total number of updates of action-value estimates plus the number of times the escape event from K_t, A_K, can occur is bounded by $\zeta(\epsilon, \delta)$ (learning complexity). Then, when $\mathcal{A}(\epsilon, \delta)$ is executed on any MDP M, it will follow a 4ϵ-optimal policy from its current state on all but

$$O\left(\frac{\zeta(\epsilon,\delta)}{\epsilon(1-\gamma)^2}\ln\frac{1}{\delta}\ln\frac{1}{\epsilon(1-\gamma)}\right)$$

timesteps, with probability at least $1 - 2\delta$.

III. Factored Rmax

Rmax is a reinforcement-learning algorithm introduced by [2] and shown to have PAC sample complexity by [3] ([2] showed it was PAC in a slightly different setting). Factored Rmax (or **f-Rmax**) is the direct generalization to factored-state MDPs [12]. Factored Rmax is model-based, in that it maintains a model M' of the underlying f-MDP, and at each step, acts according to an optimal policy of its model.

To motivate the model used by Factored Rmax, we first describe at a high level the main intuition of the algorithm. Consider a fixed state factor X_i and dependency identifier $j \in \mathcal{I}$ such that $D(s,a,X_i) = j$ for some state s and action a. Let $x_i \in \mathcal{D}(X_i)$ be any value in the domain of X_i. There exists an associated probability $P(x_i|j)$. We call the corresponding distribution, $P(\cdot|j)$ for (X_i, j), a *transition component*, which is defined formally in Section II-A. The agent doesn't have access to this distribution however, and it must be learned. The main idea behind model-based approaches for f-MDPs is to use the agent's experience to compute an approximation to the unknown distribution $P(\cdot|j)$. However, when the agent's experience is limited, the empirical distribution often produces a very poor approximation. The trick behind f-Rmax is to use the agent's experience only when there is enough of it to ensure decent accuracy, with high probability.

Let $m = (m_1, \ldots, m_n)$ be some user-defined vector of positive integers that is provided to f-Rmax as input at the beginning of a run. For each transition component (X_i, j), f-Rmax maintains a count $n(X_i, j)$ of the number of times it has taken an action a from a state s for which $D(s,a,X_i) = j$. For a given state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$, $D_{s,a}$ is the set of relevant dependency identifies $j \in \mathcal{I}$ such that $j = D(s,a,X_i)$ for some factor X_i. If the associated counts, $n(X_i, j)$, are each at least m_i, respectively, then we say that (s,a) is a *known* state-action pair and use the reward distribution and the empirical transition distribution estimate for (s,a). Otherwise, the f-Rmax agent assumes that the value of taking action a from state s is $1/(1-\gamma)$, the maximum possible value.

Intuitively, the model used by f-Rmax provides a large exploration bonus for reaching a state-action pair that is unknown (meaning it has some relevant transition component X_i that has not been experienced m_i times). This will encourage the agent to increase the counts $n(X_i, j)$, causing effective exploration of the state space until many transition components are known. After much experience, the empirical distribution is used, and the agent acts according to a near-optimal policy. We will show that if the m_i are set large enough, but still polynomial in the relevant quantities (with the number of states being replaced by the number of transition components), then factored Rmax is PAC-fMDP in the sample complexity framework.

Formally, Rmax solves the following set of equations to compute its state-action value estimates:

$$\begin{aligned} Q(s,a) &= 1/(1-\gamma), \text{if } \exists X_i,\ n(D(s,a,X_i)) < m_i \\ Q(s,a) &= R(s,a) + \gamma \sum_{s'} \hat{T}(s'|s,a) \max_{a'} Q(s',a'), \end{aligned}$$

otherwise.

IV. Analysis of Factored Rmax

The main result of this section is the following theorem, which implies that f-Rmax is PAC-fMDP.

Theorem 2: Suppose that $0 \le \epsilon < \frac{1}{1-\gamma}$ and $0 \le \delta < 1$ are two real numbers and $M = \langle \mathcal{S}, \mathsf{A}, T, \mathcal{R}, \gamma \rangle$ is any factored-state MDP with dependency function D and dependency identifiers $\mathcal{I}$. Let n be the number of state factors and $\mathcal{Q}$ be the set of transition components with $N = |\mathcal{Q}|$. There exists inputs $m = (m_1, \ldots, m_n)$ satisfying $m_i = m_i(\frac{1}{\epsilon}, \frac{1}{\delta}) = O\left(\frac{n^2(|\mathcal{D}(X_i)| + \ln(N/\delta))}{\epsilon^2(1-\gamma)^4}\right)$, such that if f-Rmax is executed on M with inputs m, then the following holds. Let $\mathcal{A}_t$ denote f-Rmax's policy at time t and s_t denote the state at time t. With probability at least $1-\delta$, $V_M^{\mathcal{A}_t}(s_t) \ge V_M^*(s_t) - \epsilon$ is true for all but

$$O\left(\frac{n^2(\Psi + N\ln(N/\delta))}{\epsilon^3(1-\gamma)^6} \ln\frac{1}{\delta} \ln\frac{1}{\epsilon(1-\gamma)}\right), \tag{5}$$

timesteps t, where $\Psi = \sum_{(X_i,j)\in\mathcal{Q}} |\mathcal{D}(X_i)|$.

Factored Rmax models the unknown environment using the certainty-equivalence method.

A. Certainty-Equivalence Model

Let $(X_i, j) \in \mathcal{Q}$ be a fixed transition component that assigns probabilities, $P(x_k|j)$, for all $x_k \in \mathcal{D}(X_i)$. The maximum likelihood estimate of these probabilities from m_i samples has the following formula:

$$\hat{P}(x_k|j) = \frac{\#\ of\ samples\ equal\ to\ x_k}{\#\ of\ samples = m_i}. \tag{6}$$

We note that a learning algorithm obtains samples for (X_i, j) whenever it takes an action a from a state s for which $(X_i, j) \in D_{s,a}$. The *empirical model* (also called the *certainty-equivalence model*) is the transition model defined by using the maximum likelihood estimates for each transition component:

$$\hat{T}(s'|s,a) = \prod_{i=1}^{n} \hat{P}(s'_i | D(s,a,X_i)). \tag{7}$$

B. Analysis Details

We utilize Theorem 1 to prove that f-Rmax is PAC-MDP. The key insight is that after an adequate number of samples have been gathered for a given transition component, the resulting empirical transition probability distribution is close to the true one, with high probability. However, transitions in a factored-state MDP involve n transition components since there are n state factors, all of whose transitions are independently computed (see Equation 1). Next, we relate accuracy in the transition model with accuracy in the transition components.

We seek to bound the L1 distance between an approximate transition distribution of the factored-state MDP to the true transition model.

Lemma 1: Let $X = \langle x(i,j) \rangle$ and $Y = \langle y(i,j) \rangle$ be any two stochastic matrices of size $m \times n$. Let $x(\cdot, j)$ denote the jth column of the matrix X, which is a discrete probability distribution over m elements. For stochastic matrix X, let P_X denote the *product distribution* of size m^n defined as $P_X(i_1, \ldots, i_n) = x(i_1, 1) \cdots x(i_m, m)$. If

$$||x(\cdot,j) - y(\cdot,j)||_1 \le 2\epsilon \ \text{ for all } j = 1, \ldots, n,$$

for $0 \le \epsilon \le 1$, then

$$\sum_{i_1=1}^{m} \cdots \sum_{i_n=1}^{m} |P_X(i_1,\ldots,i_n) - P_Y(i_1,\ldots,i_n)| \le 2 - 2(1-\epsilon)^n$$

Proof: Using the transformation $y(i,j) = x(i,j) + \alpha(i,j)$, we seek to maximize the function $f(X, \alpha(\cdot,\cdot)) := \sum_{i_1=1}^{m} \cdots \sum_{i_n=1}^{m} |x(i_1,1)\cdots x(i_n,n) - (x(i_1,1) + \alpha(i_1,1))\cdots(x(i_n,n) + \alpha(i_n,n))|$, under the lemma's constraints. Fix any column index j and consider the partial derivative with respect to the ith component, $\partial f/\partial x(i,j)$. It is clear that this derivative does not depend[2] on $x(\cdot,j)$. Suppose there is an element in the jth column, $x(i,j)$, such that $\epsilon < x(i,j) < 1-\epsilon$. Then there must be another distinct element $x(i',j)$ such that $x(i',j) > 0$. Without loss of generality, suppose that $\partial f/\partial x(i,j) \ge \partial f/\partial x(i',j)$. The value of f will not decrease if we simultaneously increase $x(i,j)$ and decrease $x(i',j)$ by as much as possible (until $x(i,j) + \alpha(i,j) = 1$ or $x(i',j) + \alpha(i',j) = 0$). By a similar argument if there are two or more nonzero elements in the jth column that add up to ϵ, then we can increase the one with largest partial derivative to ϵ and decrease the others to zero. In conclusion, we can restrict ourselves to matrices X whose columns are one of the two following forms: (1) there is one element with value $1-\epsilon$ and another one with value ϵ, or (2) there is a single element with value one and

[2] The absolute value signs can be removed by noting that there is some setting of $\alpha(\cdot,\cdot)$ that maximizes f.

the rest are zero-valued.[3] By symmetry, if column j of matrix X is of form (1) with indices i_1, i_2 such that $x(i_1, j) = 1-\epsilon$, $x(i_2, j) = \epsilon$, then column j of matrix Y is of form (2) with $\alpha(i_1, j) = \epsilon$ and $\alpha(i_2, j) = -\epsilon$, and vice versa.

We have shown that we can restrict the maximization of f over stochastic matrices X and Y of the following form:

$$X = \begin{pmatrix} 1-\epsilon & \dots & 1-\epsilon & 1 & \dots & 1 \\ \epsilon & \dots & \epsilon & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \end{pmatrix}$$

$$Y = \begin{pmatrix} 1 & \dots & 1 & 1-\epsilon & \dots & 1-\epsilon \\ 0 & \dots & 0 & \epsilon & \dots & \epsilon \\ 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \end{pmatrix}$$

Suppose there are t_1 columns of type (1) and $t_2 = n - t_1$ columns of type (2) in X. Then, we have that

$$f(X, Y) = |(1-\epsilon)^{t_1} - (1-\epsilon)^{t_2}| + (1-(1-\epsilon)^{t_1}) + (1-(1-\epsilon)^{t_2}). \tag{8}$$

The first term in Equation 8 follows from choosing elements only in the first row ($i_1 = \cdots = i_n = 1$). The second term results from choosing non-zero terms in X that are zero in Y and the third term from choosing non-zero terms in Y that are zero in X. Without loss of generality, if $t_1 > t_2$, then from Equation 8 we have that $f(X, Y) = 2 - 2(1-\epsilon)^{t_1} \leq 2 - 2(1-\epsilon)^n$, as desired. ■

Corollary 1: Let M be any factored-state MDP. Suppose that for each transition component $P(\cdot|j)$ we have an estimate $\hat{P}(\cdot|j)$ such that $||P(\cdot|j) - \hat{P}(\cdot|j)||_1 \leq \epsilon/n$. Then for all state-action pairs (s, a) we have that

$$||T(s, a) - \hat{T}(s, a)||_1 \leq \epsilon. \tag{9}$$

Proof: Follows directly from the fact that $1 - (1-\epsilon)^n \leq \epsilon n$. ■

We will make use of the following Lemma from [6] that relates accuracy in the model to accuracy in the value function.

Lemma 2: Let $M_1 = \langle \mathcal{S}, \mathsf{A}, T_1, R, \gamma \rangle$ and $M_2 = \langle \mathcal{S}, \mathsf{A}, T_2, R, \gamma \rangle$ be two MDPs. There exists a constant $C > 0$ such that if $||T_1(s, a) - T_2(s, a)||_1 \leq C\Big(\epsilon(1-\gamma)^2\Big)$ holds for all states s and actions a, then

$$|Q^{\pi}_{M_1}(s, a) - Q^{\pi}_{M_2}(s, a)| \leq \epsilon. \tag{10}$$

holds for any stationary policy π.

Corollary 1 shows that L1 accuracy of the transition components of the model implies L1 accuracy of the resulting empirical transition distributions. Lemma 2 shows that L1 accuracy in the transition distributions of a model combined with the true reward distributions can be used to compute accurate value functions for any policy. Thus, we are left with answering the question: *how many samples are needed to estimate a given transition component (discrete probability distribution) to a desired L1 accuracy*. The following theorem is helpful in this matter.

Theorem 3: (from [17]) Let P a probability distribution on the set $\mathcal{A} = \{1, 2, \dots, a\}$. Let $X_1, X_2, \dots, X_m$ be independent identically distributed random variables according to P. Let $\hat{P}$ denote the empirical distribution computed by using the X_i's. Then for all $\epsilon > 0$,

$$Pr(||P - \hat{P}||_1 \geq \epsilon) \leq 2^a e^{-m\epsilon^2/2}. \tag{11}$$

Definition 13: For f-Rmax we define the "known" state-action pairs K_t, at time t, to be

$$K_t := \{(s, a) \in \mathcal{S} \times \mathsf{A} | n(X_i, j) \geq m_i \text{ for all } (j, X_i) \in D_{s,a}\}. \tag{12}$$

If t is contextually defined, we use the simpler notation K. The following event will be used in our proof that f-Rmax is PAC-fMDP. We will provide a sufficient value of the parameter vector m to guarantee that the event occurs, with high probability. In words, the condition says that the value of any state s, under any policy, in the empirical known state-action MDP $\hat{M}_{K_t}$ (see Definition 11) is ϵ-close to its value in the true known state-action MDP M_{K_t}.

Event A1 *For all stationary policies π, timesteps t and states s during execution of the f-Rmax algorithm on some f-MDP M, $|V^{\pi}_{M_{K_t}}(s) - V^{\pi}_{\hat{M}_{K_t}}(s)| \leq \epsilon$.*

Lemma 3: There exists a constant C such that if f-Rmax with parameters $m = \langle m_i \rangle$ is executed on any MDP $M = \langle \mathcal{S}, \mathsf{A}, T, \mathcal{R}, \gamma \rangle$ and m_i, for $i = 1, \dots, n$, satisfies

$$m_i \geq C\left(\frac{n^2(|\mathcal{D}(X_i)| + \ln(N/\delta))}{{\epsilon_1}^2(1-\gamma)^4}\right),$$

then event $A1$ will occur with probability at least $1 - \delta$.

Proof: Event A1 occurs if f-Rmax maintains a close approximation of its known state-action MDP. On any fixed timestep t, the transition distributions that f-Rmax uses are the empirical estimates as described in Section IV-A, using only the first m_i samples (of immediate reward and next state pairs) for each $(X_i, j) \in \cup_{(s,a)\in K}\{D_{s,a}\}$. Intuitively, as long as each m_i is large enough, the empirical estimates for these state-action pairs will be accurate, with high probability[4]. Combining Corollary 1 with Lemma 2 reveals that it is sufficient to obtain $C\Big(\epsilon(1-\gamma)^2/n\Big)$-accurate (in L1 distance) transition components. From Theorem 3 we can guarantee the empirical transition distribution is accurate enough, with probability at least $1 - \delta'$, as long as $2^{|\mathcal{D}(X_i)|}e^{-m_i\epsilon^2(1-\gamma)^4/(2n^2)} \leq \delta'$. Using this expression, we find that it is sufficient to choose m such that

$$m_i \propto \frac{n^2(|\mathcal{D}(X_i)| + \ln(1/\delta'))}{\epsilon^2(1-\gamma)^4}. \tag{13}$$

[3] Technically, we have left open the possibility that one element has value $1 - \epsilon'$ and another has value ϵ', where $0 < \epsilon' < \epsilon$. However, it is easy to show that we can increase f in such a case.

[4] There is a minor technicality here. The samples, in the form of next state factors, experienced by an online agent in an f-MDP are not necessarily independent samples. The reason for this is that the learning environment or the agent could prevent future experiences of state factors based on previously observed outcomes. Nevertheless, all the tail inequality bounds that hold for independent samples also hold for online samples in f-MDPs, a fact that is due to the Markov property. There is an extended discussion and formal proof of this in the context of general MDPs in our forthcoming paper [8] that extends to the factored case.

Thus, as long as m_i is large enough, we can guarantee that the empirical distribution for a single transition component will be sufficiently accurate, with high probability. However, to apply the simulation bounds of Lemma 2, we require accuracy for all transition components. To ensure a total failure probability of δ, we set $\delta' = \delta/N$ in the above equations and apply the union bound over all transition components. ■

C. Proof of Main Theorem

Proof: (of Theorem 2). We apply Theorem 1. Assume that Event $A1$ occurs. Consider some fixed time t. First, we verify condition (1) of the theorem. We have that $V_t(s) = V^*_{\hat{M}_{K_t}}(s) \geq V^*_{M_{K_t}}(s) - \epsilon \geq V^*(s) - \epsilon$. The first equality follows from the fact that action-values used by f-Rmax are the result of a solution of its internal model. The first inequality follows from Event A1 and the second from the fact that M_{K_t} can be obtained from M by removing certain states and replacing them with a maximally rewarding state whose actions are self-loops, an operation that only increases the value of any state. Next, we note that condition (2) of the theorem follows from Event A1. Observe that the learning complexity, $\zeta(\epsilon, \delta)$, satisfies $\zeta(\epsilon, \delta) \leq \sum_{(X_i,j)\in\mathcal{Q}} m_i$. This is true because each time an escape occurs, some $(s, a) \notin K$ is experienced. However, once all the transition components (X_i, j) for (s, a) are experienced m_i times, respectively, (s, a) becomes part of and never leaves the set K. To guarantee that Event $A1$ occurs with probability at least $1 - \delta$, we use Lemma 4 to set m. ■

V. Factored IE

Interval Estimation (IE) is an advanced technique for handling exploration. It was introduced by [4] for use in the *k-armed bandit problem*, which involves learning in a special class of MDPs. The approach can be incorporated into a sample-efficient RL algorithm called Model-Based Interval Estimation or MBIE [5], [8]. In this section we demonstrate that the IE approach to efficient exploration can be applied to the problem of learning in factored-state MDPs.

The *Factored IE* (or *f-IE*) algorithm is similar to f-Rmax in that it maintains empirical estimates for the transition components as described in Section IV-A. The main difference is that f-IE uses the empirical estimates for each transition component even if the agent has little experience (in the form of samples) with respect to that component. Like f-Rmax, f-IE has a parameter m_i for each factor X_i and it uses only the first m_i samples for each transition component $(X_i, j) \in \mathcal{Q}$ to compute its empirical estimate (all additional observed samples are discarded)[5]. However, when m_i samples are yet to be obtained, f-IE still computes an empirical estimate $\hat{Pr}(\cdot|j)$ using all the observed samples. This is in contrast to the f-Rmax algorithm, which ignores such estimates. Thus, f-IE makes better use of the agent's limited experience.

For a specified state-action pair (s, a), let c_i denote the count $n(D(s, a, X_i))$ that is maintained by both the f-Rmax and f-IE algorithms. Recall that f-Rmax solves the following set of equations to compute the policy it follows:

$$\begin{aligned} Q(s,a) &= 1/(1-\gamma), \text{if } \exists X_i,\ c_i < m_i \\ Q(s,a) &= R(s,a) + \gamma \sum_{s'} \hat{T}(s'|s,a) \max_{a'} Q(s',a'), \end{aligned}$$

otherwise.

The algorithm f-IE solves a similar set of equations:

$$\begin{aligned} Q(s,a) &= 1/(1-\gamma), \text{if } \exists X_i,\ c_i = 0 \\ Q(s,a) &= R(s,a) + \gamma \sum_{s'} \hat{T}(s'|s,a) \max_{a'} Q(s',a') \\ &\quad + eb(c_1, c_2, \ldots, c_n) \qquad\qquad \text{otherwise.} \end{aligned}$$

where $eb : \mathbb{Z}^n \to \mathbb{R}$ is a function of the form

$$eb(c_1, c_2, \ldots, c_n) := \max_{(X_i,j)\in D(s,a)} \frac{\beta_i}{\sqrt{c_i}}, \tag{14}$$

for some constants β_i, $i = 1, \ldots, n$. We think of this function as an *exploration bonus* that provides incentive for obtaining samples from transition components that are poorly modeled and therefore have a low count, c_i.

VI. Analysis of Factored IE

The main result of this section is the following theorem.

Theorem 4: Suppose that $0 \leq \epsilon < \frac{1}{1-\gamma}$ and $0 \leq \delta < 1$ are two real numbers and $M = \langle \mathcal{S}, \mathsf{A}, T, \mathcal{R}, \gamma \rangle$ is any factored-state MDP with dependency function D and dependency identifiers $\mathcal{I}$. Let n be the number of state factors and $\mathcal{Q}$ be the set of transition components with $N = |\mathcal{Q}|$. There exists inputs $m = (m_1, \ldots, m_n)$ and $\beta = (\beta_1, \ldots, \beta_n)$, satisfying $m_i = m_i(\frac{1}{\epsilon}, \frac{1}{\delta}) = O\left(\frac{n^2(|\mathcal{D}(X_i)| + \ln(Nn/(\epsilon(1-\gamma)\delta)))}{\epsilon^2(1-\gamma)^4}\right)$ and $\beta_i = \frac{n}{1-\gamma}\sqrt{2\ln(Nm_i/\delta) + 2\ln(2)|\mathcal{D}(X_i)|}$, such that if f-IE is executed on M with inputs m and β, then the following holds. Let $\mathcal{A}_t$ denote f-IE's policy at time t and s_t denote the state at time t. With probability at least $1 - \delta$, $V^{\mathcal{A}_t}_M(s_t) \geq V^*_M(s_t) - \epsilon$ is true for all but

$$O\left(\frac{n^2(\Psi + N\ln(\frac{Nn}{\epsilon(1-\gamma)\delta}))}{\epsilon^3(1-\gamma)^6} \ln\frac{1}{\delta} \ln\frac{1}{\epsilon(1-\gamma)}\right), \tag{15}$$

timesteps t, where $\Psi = \sum_{(X_i,j)\in\mathcal{Q}} |\mathcal{D}(X_i)|$.

A. Analysis Details

Recall that for a fixed transition component (X_i, j), the f-IE algorithm maintains a count $n(X_i, j)$ that is equal to the number of samples obtained by the agent for estimation of the corresponding distribution. Since the algorithm will only use the first m_i samples, $n(X_i, j) \leq m_i$.

Event A2 *For all transition components $(X_i, j) \in \mathcal{Q}$, the following holds during execution of the f-IE algorithm on MDP M,*

$$||\hat{P}(\cdot|j) - P(\cdot|j)||_1 \leq \frac{\sqrt{2\ln(Nm_i/\delta) + 2\ln(2)|\mathcal{D}(X_i)|}}{\sqrt{n(X_i, j)}}, \tag{16}$$

[5] This condition was needed for our analysis to go through. Experimentally, we have found that the algorithm has reduced sample complexity but increased computational complexity without this restriction.

Lemma 4: The event $A2$ will occur with probability at least $1-\delta$.

Proof: Fix a transition component $(X_i, j) \in \mathcal{Q}$. Fix a moment during execution of f-IE in an f-MDP M. By Theorem 3, we have that $P(||P(\cdot|j) - \hat{P}(\cdot|j)||_1 \geq \alpha) \leq 2^{|\mathcal{D}(X_i)|} e^{-n(X_i,j)\alpha^2/2}$. Setting the right-hand side to be at most $\delta/(Nm_i)$ and solving for α proves that with probability at least $1-\delta/(Nm_i)$, we will have that

$$||\hat{P}(\cdot|j) - P(\cdot|j)||_1 \leq \frac{\sqrt{2\ln{(Nm_i/\delta)} + 2\ln(2)|\mathcal{D}(X_i)|}}{\sqrt{n(X_i,j)}}, \tag{17}$$

To guarantee that this holds for all transition components we proceed with two applications of the union bound: first for a fixed transition component over all possible values of $n(X_i, j)$ and then for a fixed factor over all transition components. Let $F(X_i, j, k)$ denote the probability that Equation 17 does not hold for some timestep such that $n(X_i, j) = k$ holds. We have that

$$\sum_{(X_i,j)\in\mathcal{Q}} \sum_{k=1}^{m_i} F(X_i, j, k) \leq \sum_{(X_i,j)\in\mathcal{Q}} \sum_{k=1}^{m_i} \delta/(Nm_i) = \delta$$

■

Lemma 5: If Event A2 occurs, then the following always holds during execution of f-IE: $||T(s,a) - \hat{T}(s,a)||_1 \leq$

$$\max_{(X_i,j)\in D(s,a)} \frac{n\sqrt{2\ln{(Nm_i/\delta)} + 2\ln(2)|\mathcal{D}(X_i)|}}{\sqrt{n(X_i,j)}}, \tag{18}$$

for all $(s,a) \in \mathcal{S}\times\mathcal{A}$.

Proof: The claim follows directly from Corollary 1. ■

Lemma 6: If Event A2 occurs and

$$\beta_i \geq \frac{n}{1-\gamma}\sqrt{2\ln{(Nm_i/\delta)} + 2\ln(2)|\mathcal{D}(X_i)|}, \tag{19}$$

then the following always holds during execution of f-IE:

$$Q(s,a) \geq Q^*(s,a) \tag{20}$$

Proof: Recall that f-IE computes it action-value estimates, $Q(s,a)$, by solving its internal model. We prove the claim by induction on the number of steps of value iteration. Let $Q^{(i)}(s,a)$ denote the result of running value iteration of f-IE's model for i iterations. We let $Q^{(0)} = 1/(1-\gamma)$. Assume that the claim holds for some value $t-1$. We have that

$$\begin{aligned}
&Q^*(s,a) - Q^{(t)}(s,a)\\
&\leq \quad Q^*(s,a) - R(s,a) - \gamma\sum_{s'}\hat{T}(s'|s,a)V^*(s')\\
&\quad\quad -eb(c_1, c_2, \ldots, c_n)\\
&\leq \quad \frac{1}{1-\gamma}\sum_{s'}(T(s'|s,a) - \hat{T}(s'|s,a))\\
&\quad\quad - \max_{(X_i,j)\in D(s,a)} \frac{\beta_i}{\sqrt{n(X_i,j)}}\\
&\leq \quad 0.
\end{aligned}$$

The first inequality results from the induction hypothesis and the fact that $Q^{(t)}(s,a) = R(s,a) + \gamma\sum_{s'}\hat{T}(s'|s,a)\max_{a'}Q^{(t-1)}(s',a')$. The second inequality follows from the fact that $V^*(s) \leq 1/(1-\gamma)$ holds for all states s. The final inequality used Lemma 5 and Equation 19. ■

B. *Proof of Main Theorem*

Proof: (of Theorem 4). We apply Theorem 1. Assume that Event $A2$ occurs. Define the set of "known" state-action pairs K_t, at time t, to be the same as for f-Rmax:

$$K_t := \{(s,a) \in \mathcal{S}\times\mathsf{A} | n(X_i,j) \geq m_i \text{ for all } (j, X_i) \in D_{s,a}\}. \tag{21}$$

Consider some fixed time t. Condition (1) of the theorem holds by Lemma 6.

Next, we sketch a proof that condition (2) of the theorem holds. f-IE computes its action-value estimates by solving the empirical model with exploration bonuses added to the reward function. We need to show that it is close to the MDP M_{K_t}, which is identical to f-IE's model except that the true transition distribution is used instead of the empirical estimate for those state-action pairs in K_t and the exploration bonuses are discarded for those state-action pairs. If each exploration bonus is less than $\epsilon(1-\gamma)/2$ then the value function of the model is $\epsilon/2$-close to the value function of the model without the exploration bonuses (because any one-step reward gets multiplied by $1/(1-\gamma)$ if accrued over an infinite horizon). Thus we need to choose m so that

$$\beta_i/m_i \leq \epsilon(1-\gamma)/2 \text{ for each } i. \tag{22}$$

Also, by Lemma 2, we can guarantee the the value functions for the two models are ϵ-accurate as long as the transition function is $C\epsilon(1-\gamma)^2$-accurate for some constant. From Lemma 5, it is sufficient to ensure that

$$\frac{n\sqrt{2\ln{(Nm_i/\delta)} + 2\ln(2)|\mathcal{D}(X_i)|}}{\sqrt{m_i}} \leq C\epsilon(1-\gamma)^2 \text{ for each } i, \tag{23}$$

holds. Ignoring constants, the conditions specified by Equations 22 and 23 are equivalent and are satisfied by

$$m_i \propto \frac{n^2(|\mathcal{D}(X_i)| + \ln{(Nn/(\epsilon(1-\gamma)\delta))})}{\epsilon^2(1-\gamma)^4}. \tag{24}$$

Finally, note that the learning complexity, $\zeta(\epsilon,\delta) \leq \sum_{(X_i,j)\in\mathcal{Q}} m_i$. This is true because each time an escape occurs, some $(s,a) \notin K$ is experienced. However, once all the transition components (X_i, j) for (s,a) are experienced m_i times, respectively, (s,a) becomes part of and never leaves the set K. ■

VII. Conclusion

We have extended and tightened the analysis of [11] to cover the Factored Rmax and Factored IE algorithms in a more general learning framework. Our analysis made several restrictive assumptions (also present in [11]):

- **The Planning Assumption** We have shown that f-Rmax and f-IE act near-optimally on all but a small (polynomial in the number of parameters of the compact representation) number of timesteps, with high probability. Unfortunately, to do so, the algorithms must solve their model completely and exactly. It is easy to extend the analysis to allow the algorithms to solve their models only ϵ-approximately. However, even an approximate solution probably requires an exponential number of computations in either the time horizon $\left(\propto \frac{\ln(1/\epsilon)}{1-\gamma}\right)$ or in the number of parameters of the compact model. There are strong complexity results proving the hardness of planning in factored domains [18]–[20].

Thus, the algorithms we've analyzed are not computationally efficient with a naive implementation and in the worst case it may be impossible to make them so. However, in applications of RL algorithms, the critical resource is often the number of available real-world experiences or samples that the agent observes after taking an action rather than the computation performed by the agent. In addition, several approximation techniques (especially linear) have been developed and shown to work in some cases (see [12] and references within). Nonetheless, our future work includes examining conditions under which the amount of computation required by algorithms for Factored MDPs can be reduced.

- **The Known-Structure Assumption** The f-Rmax and f-IE algorithms require as input the dependency function D. This amounts to being given the underlying structure of the various dependencies in the Factored MDP. In many cases, a domain expert can easily identify the structural dependencies. Then, both algorithms that we've considered can simultaneous learn the parameters of the model and how to behave near-optimally. However, if the structure is unknown, then the algorithms cannot be directly applied. Important future work is to develop methods that learn the structure along with the parameters. Some intriguing preliminary work on this can be found in [21].

ACKNOWLEDGMENT

Thanks to the National Science Foundation (IIS-0325281) for funding and to Michael Littman for suggestions.

REFERENCES

[1] M. J. Kearns and S. P. Singh, "Near-optimal reinforcement learning in polynomial time," *Machine Learning*, vol. 49, no. 2–3, pp. 209–232, 2002.

[2] R. I. Brafman and M. Tennenholtz, "R-MAX—a general polynomial time algorithm for near-optimal reinforcement learning," *Journal of Machine Learning Research*, vol. 3, pp. 213–231, 2002.

[3] S. M. Kakade, "On the sample complexity of reinforcement learning," Ph.D. dissertation, Gatsby Computational Neuroscience Unit, University College London, 2003.

[4] L. P. Kaelbling, *Learning in Embedded Systems*. Cambridge, MA: The MIT Press, 1993.

[5] M. Wiering and J. Schmidhuber, "Efficient model-based exploration," in *Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior (SAB'98)*, 1998, pp. 223–228.

[6] A. L. Strehl and M. L. Littman, "A theoretical analysis of model-based interval estimation," in *Proceedings of the Twenty-second International Conference on Machine Learning (ICML-05)*, 2005, pp. 857–864.

[7] ——, "An empirical evaluation of interval estimation for Markov decision processes," in *The 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI-2004)*, 2004, pp. 128–135.

[8] ——, "An analysis of model-based interval estimation for Markov decision processes," *Journal of Computer and System Sciences*, in press.

[9] D. Koller and R. Parr, "Computing factored value functions for policies in structured MDPs," in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*. The AAAI Press/The MIT Press, 1999, pp. 1332–1339.

[10] C. Boutilier, T. Dean, and S. Hanks, "Decision-theoretic planning: Structural assumptions and computational leverage," *Journal of Artificial Intelligence Research*, vol. 11, pp. 1–94, 1999.

[11] M. J. Kearns and D. Koller, "Efficient reinforcement learning in factored MDPs," in *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, 1999, pp. 740–747.

[12] C. Guestrin, R. Patrascu, and D. Schuurmans, "Algorithm-directed exploration for model-based reinforcement learning in factored MDPs," in *Proceedings of the International Conference on Machine Learning*, 2002, pp. 235–242.

[13] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, 1998.

[14] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller, "Context-specific independence in Bayesian networks," in *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI 96)*, Portland, OR, 1996, pp. 115–123.

[15] L. G. Valiant, "A theory of the learnable," *Communications of the ACM*, vol. 27, no. 11, pp. 1134–1142, November 1984.

[16] A. L. Strehl, L. Li, and M. L. Littman, "Incremental model-based learners with formal learning-time guarantees," in *UAI-06: Proceedings of the 22nd conference on Uncertainty in Artificial Intelligence*, 2006, pp. 485–493.

[17] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. J. Weinberger, "Inequalities for the L1 deviation of the empirical distribution," Hewlett-Packard Labs, Tech. Rep. HPL-2003-97R1, 2003.

[18] M. L. Littman, "Probabilistic propositional planning: Representations and complexity," in *Proceedings of the Fourteenth National Conference on Artificial Intelligence*. AAAI Press/The MIT Press, 1997, pp. 748–754. [Online]. Available: http://www.cs.rutgers.edu/ mlittman/papers/aaai97-planning.ps

[19] M. L. Littman, J. Goldsmith, and M. Mundhenk, "The computational complexity of probabilistic planning," *Journal of Artificial Intelligence Research*, vol. 9, pp. 1–36, 1998.

[20] E. Allender, S. Arora, M. Kearns, C. Moore, and A. Russell, "A note on the representational incompatabilty of function approximation and factored dynamics." in *Advances in Neural Information Processing Systems (NIPS-03)*, 2002.

[21] T. Degris, O. Sigaud, and P.-H. Wuillemin, "Learning the structure of factored Markov decision processes in reinforcement learning problems," in *ICML-06: Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 257–264.