Optimistic Initialization and Greediness Lead to Polynomial Time Learning in Factored MDPs

István Szita SZITYU@GMAIL.COM
Rutgers University, 110 Frelinghuysen Road, Piscataway, NJ, 08854, USA

András Lőrincz ANDRAS.LORINCZ@INF.ELTE.HU
Eötvös Loránd University, 1116 Budapest, Pázmány P. sétány 1/C, Hungary

Abstract

In this paper we propose an algorithm for polynomial-time reinforcement learning in factored Markov decision processes (FMDPs). The factored optimistic initial model (FOIM) algorithm, maintains an empirical model of the FMDP in a conventional way, and always follows a greedy policy with respect to its model. The only trick of the algorithm is that the model is initialized optimistically. We prove that with suitable initialization (i) FOIM converges to the fixed point of approximate value iteration (AVI); (ii) the number of steps when the agent makes non-near-optimal decisions (with respect to the solution of AVI) is polynomial in all relevant quantities; (iii) the per-step costs of the algorithm are also polynomial. To our best knowledge, FOIM is the first algorithm with these properties.

1. Introduction

Factored Markov decision processes (FMDPs) are practical ways to compactly formulate sequential decision problems—provided that we have ways to solve them. When the environment is unknown, all effective reinforcement learning methods apply some form of the "optimism in the face of uncertainty" principle: whenever the learning agent faces the unknown, it should assume high rewards in order to encourage exploration. *Factored optimistic initial model* (FOIM) takes this principle to the extreme: its model is initialized to be overly optimistic. For more often visited areas of the state space, the model gradually gets more realistic, inspiring the agent to head for unknown regions and explore them, in search of some imaginary "Garden of Eden". The working of the algorithm is simple to the extreme: it will not make any explicit effort to balance exploration and exploitation, but always follows the greedy optimal policy with respect to its model. We show in this paper that this simple (even simplistic) trick is sufficient for effective FMDP learning.

The algorithm is an extension of OIM (*optimistic initial model*) (Szita & Lőrincz, 2008b), which is a sample-efficient learning algorithm for flat MDPs. There is an important difference, however, in the way the model is solved. Every time the model is updated, the corresponding value function needs to be re-calculated (or updated) For flat MDPs, this is not a problem: various dynamic programming-based algorithms (like value iteration) can solve the model to any required accuracy in polynomial time.

The situation is less bright for generating near-optimal FMDP solutions: all currently known algorithms may take exponential time, e.g. the approximate policy iteration of Boutilier et al. (2000) using decision-tree representations of policies, or solving the exponential-size flattened version of the FMDP. If we require polynomial running time (as we do in this paper in search for a practical algorithm), then we have to accept sub-optimal solutions (Liberatore, 2002). The only known example of a polynomial-time FMDP planner is *factored value iteration* (FVI) (Szita & Lőrincz, 2008a), which will serve as the base planner for our learning method. This planner is guaranteed to converge, and the error of its solution is bounded by a term depending only on the quality of function approximators.

Our analysis of the algorithm will follow the established techniques for analyzing sample-efficient rein-

Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009. Copyright 2009 by the author(s)/owner(s).

forcement learning (like the works of Kearns and Singh (1998); Brafman and Tennenholtz (2001); Strehl and Littman (2005); Szita and Lőrincz (2008b) on flat MDPs and Strehl (2007) on FMDPs). However, the listed proofs of convergence rely critically on access to a near-optimal planner, so they have to be generalized suitably. By doing so, we are able to show that FOIM converges to a bounded-error solution in polynomial time with high probability.

We introduce basic concepts and notations in section 2, then in section 3 we review existing work, with special emphasis to the immediate ancestors of our method. In sections 4 and 5 we describe the blocks of FOIM and the FOIM algorithm, respectively. We finish the paper with a short analysis and discussion.

2. Basic concepts and notations

An MDP is characterized by a quintuple $(\mathbf{X}, A, R, P, \gamma)$, where $\mathbf{X}$ is a finite set of states; A is a finite set of possible actions; $R : \mathbf{X} \times A \to \mathbb{R}$ is the reward function of the agent; $P : \mathbf{X} \times A \times \mathbf{X} \to [0, 1]$ is the transition function; and finally, $\gamma \in [0, 1)$ is the discount rate on future rewards. A (stationary, Markov) policy of the agent is a mapping $\pi : \mathbf{X} \times A \to [0, 1]$. The optimal value function $V^* : \mathbf{X} \to \mathbb{R}$ gives the maximum attainable total rewards for each state, and satisfies the Bellman equation

$$V^*(\mathbf{x}) = \max_a \sum_{\mathbf{y}} P(\mathbf{y} \mid \mathbf{x}, a)\Big(R(\mathbf{x}, a) + \gamma V^*(\mathbf{y})\Big). \quad (1)$$

Given the optimal value function, it is easy to get an optimal policy: $\pi^*(\mathbf{x}, a) := 1$ iff $a = \arg\max_a \sum_{\mathbf{y}} P(\mathbf{y} \mid \mathbf{x}, a)\Big(R(\mathbf{x}, a) + \gamma V^*(\mathbf{y})\Big)$ and 0 otherwise.

2.1. Vector notation

Let $N := |\mathbf{X}|$, and suppose that states are integers from 1 to N, i.e. $\mathbf{X} = \{1, 2, \ldots, N\}$. Clearly, value functions are equivalent to N-dimensional vectors of reals, which may be indexed with states. The vector corresponding to V will be denoted as $\mathbf{v}$ and the value of state $\mathbf{x}$ by $\mathbf{v}_{\mathbf{x}}$. Similarly, for each a let us define the N-dimensional column vector $\mathbf{r}^a$ with entries $\mathbf{r}^a_{\mathbf{x}} = R(\mathbf{x}, a)$ and $N \times N$ matrix P^a with entries $P^a_{\mathbf{x},\mathbf{y}} = P(\mathbf{y} \mid \mathbf{x}, a)$.

The Bellman equations can be expressed in vector notation as $\mathbf{v}^* = \mathbf{max}_{a \in A}\big(\mathbf{r}^a + \gamma P^a \mathbf{v}^*\big)$, where $\mathbf{max}$ denotes the componentwise maximum operator. The Bellman equations are the basis to many RL algorithms, most notably, value iteration:

$$\mathbf{v}_{t+1} := \mathbf{max}_{a \in A}\big(\mathbf{r}^a + \gamma P^a \mathbf{v}_t\big), \quad (2)$$

which converges to $\mathbf{v}^*$ for any initial vector $\mathbf{v}_0$.

2.2. Factored structure

We assume that $\mathbf{X}$ is the Cartesian product of m smaller state spaces (corresponding to individual variables):

$$\mathbf{X} = X_1 \times X_2 \times \ldots \times X_m.$$

For the sake of notational convenience we will assume that each X_i has the same size, $|X_1| = |X_2| = \ldots = |X_m| = n$. With this notation, the size of the full state space is $N = |\mathbf{X}| = n^m$. We note that all derivations and proofs carry through to different size variable spaces.

Definition 2.1 *For any subset of variable indices $Z \subseteq \{1, 2, \ldots, m\}$, let $\mathbf{X}[Z] := \times_{i \in Z} X_i$, furthermore, for any $\mathbf{x} \in \mathbf{X}$, let $\mathbf{x}[Z]$ denote the value of the variables with indices in Z. We shall also use the notation $\mathbf{x}[Z]$ without specifying a full vector of values $\mathbf{x}$, in such cases $\mathbf{x}[Z]$ denotes an element in $\mathbf{X}[Z]$. For single-element sets $Z = \{i\}$ we shall also use the shorthand $\mathbf{x}[\{i\}] = \mathbf{x}[i]$.*

Definition 2.2 (Local-scope function) *A function f is a* local-scope *function if it is defined over a subspace $\mathbf{X}[Z]$ of the state space, where Z is a (presumably small) index set.*

If $|Z|$ is small, local-scope functions can be represented efficiently, as they can take only $n^{|Z|}$ different values.

Definition 2.3 (Extension) *For $f : \mathbf{X}[Z] \to \mathbb{R}$ be a local-scope function. Its extension to the whole state space is defined by $f(\mathbf{x}) := f(\mathbf{x}[Z])$. The extension operator for Z is a linear operator with a matrix $E_{[Z]} \in \mathbb{R}^{|\mathbf{X}| \times |\mathbf{X}[Z]|}$, with entries*

$$\big(E_{[Z]}\big)_{\mathbf{u},\mathbf{v}[Z]} = \begin{cases} 1, & \textit{if } \mathbf{u}[Z] = \mathbf{v}[Z]; \\ 0, & \textit{otherwise.} \end{cases}$$

For any local-scope function f with a corresponding vector representation $\mathbf{f} \in \mathbb{R}^{|\mathbf{X}[Z]| \times 1}$, $E_{[Z]}\mathbf{f} \in \mathbb{R}^{|\mathbf{X}| \times 1}$ is the vector representation of the extended function.

We assume that the reward function is the sum of J local-scope functions with scopes Z_j: $R(\mathbf{x}, a) = \sum_{j=1}^J R_j(\mathbf{x}[Z_j], a)$. In vector notation: $\mathbf{r}^a = \sum_{j=1}^J E_{[Z^a_j]}\mathbf{r}^a_i$. We also assume that for each variable i there exist neighborhood sets Γ_i such that the value of $\mathbf{x}_{t+1}[i]$ depends only on $\mathbf{x}_t[\Gamma_i]$ and the action a_t taken. Then we can write the transition probabilities in a factored form

$$P(\mathbf{y} \mid \mathbf{x}, a) = \prod_{i=1}^{m} P_i(\mathbf{y}[i] \mid \mathbf{x}[\Gamma_i], a) \tag{3}$$

for each $\mathbf{x}, \mathbf{y} \in \mathbf{X}$, $a \in A$, where each factor is a local-scope function $P_i : \mathbf{X}[\Gamma_i] \times A \times X_i \to [0,1]$ (for all $i \in \{1, \ldots, m\}$). In vector/matrix notation, for any vector $\mathbf{v} \in \mathbb{R}^{|\mathbf{X}|\times 1}$, $P^a\mathbf{v} = \bigotimes_{i=1}^{m}(P_i^a \mathbf{v}[\Gamma_i])$, where $\bigotimes$ denotes the Kronecker product. Finally, we assume that the size of all local scopes are bounded by a small constant $m_f \ll m$: $|\Gamma_i| \le m_f$ for all i. As a consequence, all probability factors can be represented with tables having at most $N_f := n^{m_f}$ rows.

An FMDP is fully characterized by the tuple $\mathcal{M} = \big(\{X_i\}_{i=1}^m; A; \{Z_j\}_{j=1}^J; \{R_j\}_{j=1}^J; \{\Gamma_i\}_{i=1}^m; \{P_i\}_{i=1}^m; \mathbf{x}_s; \gamma\big)$.

3. Related literature

The idea of representing a large MDP using a factored model was first proposed by Koller and Parr (2000) but similar ideas appear already in the works of Boutilier et al. (1995); Boutilier et al. (2000).

3.1. Planning in known FMDPs

Decision trees (or equivalently, decision lists) provide a way to represent the agent's policy compactly. Koller and Parr (2000) and Boutilier et al. (1995); Boutilier et al. (2000) present algorithms to evaluate and improve such policies, according to the policy iteration scheme. Unfortunately, the size of the policies may grow exponentially even with a decision tree representation (Boutilier et al., 2000; Liberatore, 2002).

The exact Bellman equations (1) can be transformed to an equivalent linear program with N variables and $N \cdot |A|$ constraints. In the approximate linear programming approach, we approximate the value function as a linear combination of K basis functions, resulting in an approximate LP with K variables and $N \cdot |A|$ constraints. Both the objective function and the constraints can be written in compact forms, exploiting the local-scope property of the appearing functions. Guestrin et al. (2002) show that the maximum of exponentially many local-scope functions can be computed by rephrasing the task as a non-serial dynamic programming task and eliminating variables one by one. Therefore, the equations can be transformed to an equivalent, more compact linear program. The gain may be exponential, but this is not necessarily so in all cases. Furthermore, solutions will not be (near-)optimal because of the function approximation; the best that can be proved is bounded error from the optimum (where the bound depends on the quality of basis functions used for approximation).

The approximate policy iteration algorithm (Koller & Parr, 2000; Guestrin et al., 2002) also uses an approximate LP reformulation, but it is based on the policy-evaluation Bellman equations. Policy-evaluation equations are, however, linear and do not contain the maximum operator, so there is no need for a costly transformation step. On the other hand, the algorithm needs an explicit decision tree representation of the policy. Liberatore (2002) has shown that the size of the decision tree representation can grow exponentially. Furthermore, the convergence properties of these algorithms are unknown.

Factored value iteration (Szita & Lőrincz, 2008a) also approximates the value function as a linear combination of basis functions, but uses a variant of approximate value iteration: the projection operator is modified to avoid divergence. FVI converges in a polynomial number of steps, but the solution may be sub-optimal. The error of the solution has bounded distance from the optimal value function, where the bound depends on the quality of function approximation. As an integral part of FOIM, FVI is described in detail in Section 4.1.

3.2. Reinforcement Learning in FMDPs

In the reinforcement learning setting, the agent interacts with an FMDP environment with unknown parameters. In the model-based approach, the agent has to learn the structure of the FMDP (i.e., the dependency sets Γ_i and the reward domains Z_j), the transition probability factors P_i and the reward factors R_j.

Unknown transitions. Most approaches assume that the structure of the FMDP and the reward functions are known, so only transition probabilities need to be learnt. Examples include the factored versions of sample-efficient model-based RL algorithms: factored E^3 (Kearns & Koller, 1999), factored R-max, or factored MBIE (Strehl, 2007). All the abovementioned algorithms have polynomial sample complexity (in all relevant task parameters), and require polynomially many calls to an FMDP-planner. Note however, that all of the mentioned approaches require access to a planner that is able to produce ϵ-optimal solutions[1] – and to date, no algorithm exists that would accomplish

[1]The assumption of (Kearns & Koller, 1999) is slightly less restrictive: they only require that the value of the returned policy has value at least ρV^* with some $\rho < 1$. However, no planner is known that can achieve this and cannot achieve near-optimality.

this accuracy in polynomial time. (Guestrin et al., 2002) also present an algorithm where exploration is guided by the uncertainties of the linear programming solution. While this approach does not require access to a near-optimal planner, no formal performance bounds are known.

Unknown rewards. Typically, it is asserted that the rewards can be approximated from observations analogously to transition probabilities. However, if the reward is composed of multiple factors (i.e., $J > 1$), then we can only observe the *sums* of unknown quantities, not the individual quantities themselves. To date, we know of no efficient approximation method for learning factored rewards.

Unknown structure. Few attempts exist that try to obtain the structure of the FMDP automatically. Strehl et al. (2007) present a method that learns the structure of an FMDP in polynomial time (in all relevant parameters).

4. Building blocks of FOIM

We describe the two main building blocks of our algorithm, *factored value iteration* and *optimistic initial model.*

4.1. Factored value iteration

We assume that all value functions are approximated as the linear combination of K basis functions $h_k : \mathbf{X} \to \mathbb{R}$: $V(\mathbf{x}) = \sum_{k=1}^{K} w_k h_k(\mathbf{x})$.

Let H be the $N \times K$ matrix mapping feature weights to state values, with entries $H_{\mathbf{x},k} = h_k(\mathbf{x})$, and let G be an arbitrary $K \times N$ linear mapping projecting state values to feature weights. Let $\mathbf{w} \in \mathbb{R}^K$ denote the weight vector of the basis functions. It is known (Szita & Lőrincz, 2008a) that if $\|HG\|_\infty \le 1$, then the approximate Bellman equations $\mathbf{w}^\times = G\mathbf{max}_{a\in A}\big(\mathbf{r}^a + \gamma P^a H \mathbf{w}^\times\big)$ have a unique fixed point solution $\mathbf{w}^\times$, and *approximate value iteration (AVI)*

$$\mathbf{w}_{t+1} := G\mathbf{max}_{a\in A}\big(\mathbf{r}^a + \gamma P^a H \mathbf{w}_t\big) \tag{4}$$

converges there for any starting vector $\mathbf{w}_0$.

Definition 4.1 *Let the AVI-optimal value function be defined as* $\mathbf{v}^\times = H\mathbf{w}^\times$.

As shown by Szita and Lőrincz (2008a), the distance of AVI-optimal value function from the true optimum is bounded by the projection error of $\mathbf{v}^*$:

$$\left\|\mathbf{v}^\times - \mathbf{v}^*\right\|_\infty \le \tfrac{1}{1-\gamma}\left\|HG\mathbf{v}^* - \mathbf{v}^*\right\|_\infty. \tag{5}$$

We make the further assumption that all the basis functions are local-scope ones: for each $k \in \{1,\dots,K\}$, $h_k : \mathbf{X}[C_k] \to \mathbb{R}$, with feature matrices $H_k \in \mathbb{R}^{|\mathbf{X}[C_k]|\times K}$. The feature matrix H can be decomposed as $H = \sum_{k=1}^{K} E_{[C_k]} H_k$.

Definition 4.2 *For any matrices H and G, let the row-normalization of G be a matrix $\mathcal{N}(G)$ of the same size as G, and having the entries* $[\mathcal{N}(G)]_{k,\mathbf{x}} = \frac{G_{k,\mathbf{x}}}{\|[HG]_{k,*}\|_\infty}$.

Throughout the paper, we shall use the projection matrix $G = \mathcal{N}(H^T)$.

The AVI equation (4) can be considered as the product of the $K \times N$ matrix G and an $N \times 1$ vector $\mathbf{v}_t = \mathbf{max}_{a\in A}\big(\mathbf{r}^a + \gamma P^a H\mathbf{w}_t\big)$. Using the above assumptions and notations, we can see that for any $\mathbf{x} \in \mathbf{X}$, the corresponding column of G and the corresponding element of $\mathbf{v}_t$ can be computed in polynomial time:

$$[G]_{k,\mathbf{x}} = \frac{1}{\|[HH^T]_{k,*}\|_\infty} \sum_{k'=1}^{K} \left[H_{k'}^T\right]_{*,\mathbf{x}[C_{k'}]};$$

$$[\mathbf{v}_t]_{\mathbf{x}} = \max_{a\in A}\Big[\sum_{j=1}^{J}[\mathbf{r}_j^a]_{\mathbf{x}[Z_j^a]} + \gamma \sum_{k=1}^{K} E_{[\Gamma\cup C_k]}\Big(\bigotimes_{i\in C_k} P_i^a\Big)(\mathbf{h}_k w_{k,t})\Big]$$

Factored value iteration draws $N_1 \ll N$ states uniformly at random, and performs approximate value iteration on this reduced state set.

Theorem 4.3 (Szita & Lőrincz, 2008a) *Suppose that $G = \mathcal{N}(H^T)$ For any $\epsilon > 0$, $\delta > 0$, if the sample size is $N_1 = O(\frac{m^2}{\epsilon^2}\log\frac{m}{\delta})$, then with probability at least $1-\delta$, factored value iteration converges to a weight vector $\mathbf{w}$ such that $\|\mathbf{w} - \mathbf{w}^\times\|_\infty \le \epsilon$. In terms of the optimal value function,*

$$\left\|\mathbf{v}^\times - \mathbf{v}^*\right\|_\infty \le \tfrac{1}{1-\gamma}\left\|HG\mathbf{v}^* - \mathbf{v}^*\right\|_\infty + \epsilon. \tag{6}$$

4.2. Optimistic initial model for flat MDPs

There are a number of sample-efficient learning algorithms for MDPs, e.g., E3, Rmax, MBIE, and most recently, OIM. The underlying principle of all these methods is similar: they all maintain an approximate MDP model of the environment. Wherever the uncertainty of the model parameters is high, the models are optimistic. This way, the agent is encouraged to explore the unknown areas, reducing the uncertainty of the models.

Here, we shall use and extend OIM to factored environments. In the OIM algorithm, we introduce a hypothetical "garden of Eden" (GOE) state x_E, where the agent gets a very large reward R_{E} and remains there indefinitely. The model is initialized with fake experience, according to which the agent has experienced an (x, a, x_E) transition for all $x \in X$ and $a \in A$. According to this initial model, each state has value $R_{\mathrm{E}}/(1-\gamma)$, which is a major overestimation of the true values. The model is continuously updated by the collected experience of the agent, who always takes the greedy optimal action with respect to its current model. For well-explored (x, a) pairs, the optimism of the model vanishes, thus encouraging the agent to explore the less-known areas.

The reason for choosing OIM is twofold: (1) The optimism of the model is ensured at initialization time, and after that, no extra work is needed to ensure the optimism of the model or to encourage exploration. (2) Results on several standard benchmark MDPs indicate that OIM is superior to the other algorithms mentioned.

5. Learning in FMDPs with an Optimistic initial model

Similarly to other approaches, we will make the assumptions that (a) the dependencies are known, and (b) the reward function is known, only the transition probabilities need to be learned.

5.1. Optimistic initial model for factored MDPs

During the learning process, we will maintain approximations of the model, in particular, of the transition probability factors. We extend all state factors with the hypothetical "garden of Eden" state x_E. Seeing the current state $\mathbf{x}$ and the action a taken, the transition model should give the probabilities of various next states $\mathbf{y}$. Specifically, the ith factor of the transition model should give the probabilities of various y_i values, given $\mathbf{x}[\Gamma_i]$ and a. Initially, the agent has no idea, so we let it start with an overly optimistic model: we inject the fake experience to the model that taking action a in $\mathbf{x}[\Gamma_i]$ leads to a state with ith component $y_i = x_E$. This optimistic model will encourage the agent to explore action a whenever its state is consistent with $\mathbf{x}[\Gamma_i]$. After many visits to $(\mathbf{x}[\Gamma_i], a)$, the weight of the initial fake experience will shrink, and the optimistic belief of the agent (together with its exploration-boosting effect) fades away. However, by that time, the collected experience provides an accurate approximation of the $P_i(y_i \mid \mathbf{x}[\Gamma_i], a)$ values.

So, according to the initial model (based purely on fake experience),

$$\widehat{P}(\mathbf{y} \mid \mathbf{x}, a) = \begin{cases} 1, & \text{if } \mathbf{y} = (x_E, \ldots, x_E); \\ 0, & \text{otherwise,} \end{cases}$$

$\widehat{R}(\mathbf{x}, a) = c \cdot R_{\mathrm{E}}$, if c components of $\mathbf{x}$ are x_E. This model is optimistic indeed, all non-GOE states have value at least $\gamma R_{\mathrm{E}}/(1-\gamma)$. Note that it is not possible to encode the R_{E}-rewards for the *GOE* states using the original set of reward factors, so for all state factor i, we add a new reward factor with local scope X_i: $R'_i : X_i \times A \to \mathbb{R}$, defining $R'_i(x, a) = \begin{cases} R_{\mathrm{E}}, & \text{if } x = x_E; \\ 0, & \text{otherwise.} \end{cases}$ With this modification, we are able to fully specify our algorithm, as shown in the pseudocode below.

Algorithm 1 Factored optimistic initial model.

```
input:
  $\mathcal{M} = \left(\{X_i\}_1^m; A; \{Z_j\}_1^J; \{R_j\}_1^J; \{\Gamma_i\}_1^m; \{P_i\}_1^m; \mathbf{x}_s; \gamma\right)$
  $\{H_k\}_1^K$; $\{C_k\}_1^K$; $\epsilon > 0$; $\delta > 0$; $R_{\mathrm{E}}$
initialization:
  $t := 0$;
  for all $i$, add GOE states: $X_i := X_i \cup \{x_E\}$
  for all $i$, add GOE reward function $\mathbf{r}'_i$
  for all $i$, $a$, $\mathbf{x}[\Gamma_i]$, $y \in X_i \setminus \{x_E\}$, let
  $\mathit{TransitionCount}(\mathbf{x}[\Gamma_i], a, y) := 0$;
  $\mathit{TransitionCount}(\mathbf{x}[\Gamma_i], a, x_E) := 1$;
  $\mathit{VisitCount}(\mathbf{x}[\Gamma_i], a) := 1$;
repeat
  $[\widehat{P}_i^a]_{\mathbf{x}[\Gamma_i], y} := \frac{\mathit{TransitionCount}_i(\mathbf{x}[\Gamma_i], a, y)}{\mathit{VisitCount}_i(\mathbf{x}[\Gamma_i], a)}$.
  $\mathbf{w}_t := \mathit{FactoredValueIteration}(\widehat{\mathcal{M}}, \{\widehat{P}_i^a\}, \epsilon, \delta)$
  update TransitionCount and VisitCount corresponding to transition $(\mathbf{x}_t, a_t, \mathbf{x}_{t+1})$.
  $t := t + 1$
until interaction lasts
```

5.2. Analysis

Below we prove that FOIM gets as good as possible. What is "as good as possible"? We clearly cannot expect better policies than the one the planner would output, were the parameters of the FMDP known. And because of the polynomial-running-time constraint on the planner, it will not be able to compute a near-optimal solution. However, we can prove that FOIM gets ϵ-close to the solution of the planner (which is AVI-near-optimal if the planner is FVI), except for a polynomial number of mistakes during its run.[2]

Theorem 5.1 *Suppose that an agent is following*

[2]We are using the term *polynomial* and *polynomial in all relevant quantities* as a shorthand for *polynomial in m, N_f, $|A|$, $R_{\max}$, $1/(1-\gamma)$, $1/\epsilon$ and $1/\delta$.*

FOIM in an unknown FMDP, where all reward components fall into the interval $[0, R_{\max}]$, there are m state factors, and all probability- and reward-factors depend on at most m_f factors. Let $N_f = n^{m_f}$ and let $\epsilon > 0$ and $\delta > 0$. If the initial values of FOIM satisfy

$$R_E = c \cdot \tfrac{mR_{\max}^2}{(1-\gamma)^4\epsilon}\left[\log \tfrac{mN_f|A|}{(1-\gamma)\epsilon\delta}\right],$$

then the number of timesteps when FOIM makes non-AVI-near-optimal moves, i.e., when $Q^{FOIM}(\mathbf{x}_t, a_t) < Q^{\times}(\mathbf{x}_t, a_t) - \epsilon$, is bounded by

$$O\left(\tfrac{R_{\max}^2 m^4 N_f|A|}{\epsilon^4(1-\gamma)^4}\log^3 \tfrac{1}{\delta}\log^2 \tfrac{mN_f|A|}{\epsilon}\right)$$

with probability at least $1-\delta$.

Proof sketch. The proof uses standard techniques from the literature of sample-efficient reinforcement learning. Most notably, our proof follows the structure of Strehl (2007). There are two important differences compared to previous approaches: (1) we may not assume that the planner is able to output a near-optimal solution, and (2) FOIM may make an unbounded number of model updates, so we cannot make use of the standard argument that "we are encountering only finitely many different models, each of them fails with negligible probability, so the whole algorithm fails with negligible probability". Instead, a more careful analysis of the failure probability is needed. The rigorous proof can be found in the extended version of our paper (Szita & Lőrincz, 2009)

5.2.1. Boundedness of value functions

According to our assumptions, all rewards fall between 0 and $R_{\max}$. From this, it is easy to derive an upper bound on the magnitude of the AVI-optimal value function $\mathbf{v}^{\times}$. The bound we get is $\|\mathbf{v}^{\times}\|_\infty \le \frac{3-\gamma}{1-\gamma}V_{\max} := V_0$. For future reference, we note that $V_0 = \Theta(\frac{R_{\max}}{(1-\gamma)^2})$.

5.2.2. From visit counts to model accuracy

The FOIM algorithm builds a transition probability model by keeping track of visit counts to state-action components $(\mathbf{x}[\Gamma_i], a)$ and state-action-state transition components $(\mathbf{x}[\Gamma_i], a, y)$. First of all, we show that if a state-action component is visited many times, then the corresponding probability components $\widehat{P}_{t,i}(y|\mathbf{x}[\Gamma_i], a)$ become accurate.

Let us fix a timestep $t \in \mathbb{N}$, a probability factor $i \in \{1, \dots, m\}$ and a state-action component $(\mathbf{x}[\Gamma_i], a) \in \mathbf{X}[\Gamma_i] \times A$, and $\epsilon_t > 0$. Let us denote the number of visits to the component up to time t by $k_t(\mathbf{x}[\Gamma_i], a)$. Let us introduce the shorthands $p_i = P_i(y|\mathbf{x}[\Gamma_i], a)$ and $\widehat{p}_{t,i} = \widehat{P}_{t,i}(y|\mathbf{x}[\Gamma_i], a)$. By Theorem 3 of Strehl (2007) (an application of the Hoeffding–Azuma inequality),

$$\Pr\Big(\sum_{y\in X_i}|p_i(-\widehat{p}_{t,i}|>\epsilon_t\Big) \le 2^n\exp\Big(-\tfrac{\epsilon_t^2 k_t(\mathbf{x}[\Gamma_i],a)}{2}\Big). \quad (7)$$

Unfortunately, the above inequality only speaks about a single time step t, but we need to estimate the failure probability for the whole run of the algorithm. By the union bound, that is at most

$$\sum_{k=1}^{\infty}\Pr\Big(\sum_{y\in X_i}|p_i - \widehat{p}_{t_k,i}| > \epsilon_{t_k}\Big). \quad (8)$$

Let $k_0 := \Theta\left(\frac{m^2}{(1-\gamma)^2\epsilon^2}\log\frac{m^2N_f|A|}{(1-\gamma)\delta\epsilon}\right)$. For $k < k_0$, the number of visits is too low, so in eq. (7), either ϵ_1, or the right-hand side is too big. We choose the former: we make the failure probability less than some constant δ' by setting $\epsilon_{t_k} = \frac{\beta(\delta')}{\sqrt{k}}$, where $\beta(\delta') = \sqrt{2(\log\frac{1}{\delta'} + n\log 2)}$. For $k \ge k_0$, the number of visits is sufficiently large, so we can decrease either the accuracy or the failure probability (or even both). It turns out that an approximation accuracy $\epsilon_{t_k} = \epsilon(1-\gamma)/m$ is sufficient, so we decrease failure probability. Let us set $\delta' := \Theta\left(\frac{\delta\epsilon^2(1-\gamma)^2}{m^3N_f|A|}/\log\frac{m^2N_f|A|}{(1-\gamma)\delta\epsilon}\right)$. With this choice of δ' and k_0, $\beta(\delta') \le \epsilon(1-\gamma)/m$ whenever $k \ge k_0$, furthermore, $2^n\exp\left(-\frac{k\epsilon^2}{2m^2}\right) \le \delta'$, so we get that

$$\begin{aligned}
&\sum_{k=1}^{\infty}\Pr\Big(\sum_{y\in X_i}|p_i - \widehat{p}_{t_k,i}| > \max(\tfrac{\beta(\delta')}{\sqrt{k}}, \tfrac{\epsilon(1-\gamma)}{m})\Big)\\
&\le \sum_{k=1}^{k_0-1}\delta' + \sum_{k=k_0}^{\infty}2^n\exp\left(-\tfrac{k\epsilon^2}{2m^2}\right)\\
&\le \delta'\Big(k_0 + \tfrac{1}{1-\exp\left(-\frac{\epsilon^2}{2m^2}\right)}\Big) \le \Theta\left(\tfrac{\delta}{mN_f|A|}\right).
\end{aligned}$$

We can repeat this estimation for every state-action components $(\mathbf{x}[\Gamma_i], a)$. There are at most $mN_f|A|$ of these, so the total failure probability is still less than $\Theta(\delta)$. This means that

$$\sum_{y\in X_i}|p_i - \widehat{p}_{t,i}| \le \max(\frac{\beta(\delta')}{\sqrt{k_t(\mathbf{x}[\Gamma_i], a)}}, \frac{\epsilon(1-\gamma)}{m}) \quad (9)$$

will hold for *all* $(\mathbf{x}[\Gamma_i], a)$ pairs and *all* timesteps t with high probability. From now on, we will consider only realizations where the failure event does not happen, but bear in mind that all our statements that are based on (9) are true only with $1-\Theta(\delta)$ probability.

From (9), we can easily get L_1 bounds on the accuracy of the full transition probability function: $\sum_{\mathbf{y}\in\mathbf{X}}\left|P(\mathbf{y}|\mathbf{x}, a) - \widehat{P}_t(\mathbf{y}|\mathbf{x}, a)\right| \le$

$\sum_{i=1}^{m} \max(\frac{\beta(\delta')}{\sqrt{k_t(\mathbf{x}[\Gamma_i],a)}}, \frac{\epsilon(1-\gamma)}{m})$ for all $(\mathbf{x},a) \in \mathbf{X} \times A$ and for all t.

5.2.3. The known-state FMDP

A state-action component $(\mathbf{x}[\Gamma_i],a)$ is called *known* at timestep t if it has been visited at least k_0 times, i.e., if $k_t(\mathbf{x}[\Gamma_i],a) \le k_0$. We define the *known-component* FMDP M^{K_t} as follows: (1) its state and action space, rewards, and the decompositions of the transition probabilities (i.e., the dependency sets Γ_i) are identical to the corresponding quantities of the true FMDP M, and hence to the current approximate FMDP $\widehat{M}_t$; (2) for all $a \in A$, $i \in \{1,\dots,m\}$ and $\mathbf{x}[\Gamma_i^a] \in \mathbf{X}[\Gamma_i^a]$, for any $y_i \in X_i$, the corresponding transition probability component is $P^K_{t,i}(y_i|\mathbf{x}[\Gamma_i],a) :=$

$$\begin{cases} \widehat{P}_{t,i}(y_i|\mathbf{x}[\Gamma_i],a), & \text{if } (\mathbf{x}[\Gamma_i],a) \in K_t; \\ P_i(y_i|\mathbf{x}[\Gamma_i],a), & \text{if } (\mathbf{x}[\Gamma_i],a) \notin K_t. \end{cases}$$

Note that FMDPs M^{K_t} and $\widehat{M}_t$ are very close to each other: unknown state-action components have identical transition functions by definition, while for known components, $\sum_{y \in X_i} \left| P^K_{t,i}(y|\mathbf{x}[\Gamma_i],a) - \widehat{P}_{t,i}(y|\mathbf{x}[\Gamma_i],a) \right| \le \frac{\epsilon(1-\gamma)}{m\gamma V_0}$. Consequently, for all $(\mathbf{x},a)$,

$$\sum_{\mathbf{y}\in\mathbf{X}} \left| P^K_t(\mathbf{y}|\mathbf{x},a) - \widehat{P}_t(\mathbf{y}|\mathbf{x},a) \right| \le \frac{\epsilon(1-\gamma)}{\gamma V_0}. \qquad (10)$$

For an arbitrary policy π, let $\mathbf{v}^K_\pi$ and $\widehat{\mathbf{v}}_\pi$ be the value functions (the fixed points of the approximate Bellman equations) of π in M^{K_t} and $\widehat{M}_t$, respectively. By a suitable variant of the Simulation Lemma (see supplementary material) that works with the approximate Bellman equations, we get that whenever (10) holds, $\left\| \mathbf{v}^K_\pi - \widehat{\mathbf{v}}_\pi \right\|_\infty \le \epsilon$.

5.2.4. The FOIM model is optimistic

First of all, note that FOIM is not directly using the empirical transition probabilities $\widehat{P}_{t,i}$, but it is more optimistic; it gives some chance for getting to the garden of Eden state x_E: $\widehat{P}^{FOIM}_{t,i}(y_i|\mathbf{x}[\Gamma_i],a) =$

$$\begin{cases} \frac{k_{t,i}}{k_{t,i}+1}\widehat{P}_{t,i}(y_i|\mathbf{x}[\Gamma_i],a), & \text{if } y_i \neq x_E; \\ \frac{1}{k_{t,i}+1}, & \text{otherwise,} \end{cases}$$

where we introduced the shorthand $k_{t,i} = k_t(\mathbf{x}[\Gamma_i],a)$.

Now, we show that

$$Q^\times(\mathbf{x},a) - \Big[R(\mathbf{x},a) + \gamma \sum_{\mathbf{y}\in X} \widehat{P}^{FOIM}_t(\mathbf{y}|\mathbf{x},a) V^\times(\mathbf{y}) \Big] \le \Theta(\epsilon(1-\gamma)), \qquad (11)$$

or equivalently,

$$\sum_{\mathbf{y}\in X} (P(\mathbf{y} \mid \mathbf{x},a) - P^{FOIM}_t(\mathbf{y} \mid \mathbf{x},a)) V^\times(\mathbf{y}) \ge -\sum_{i=1}^{m} \max(\tfrac{\beta}{\sqrt{k_{t,i}}}, \tfrac{\epsilon(1-\gamma)}{m}) \cdot \tfrac{k_{t,i}+1}{k_{t,i}} V_0 + \tfrac{1}{k_{t,i}} V_E.$$

Every term in the right-hand side is larger than $-\epsilon(1-\gamma)/m$, provided that we can prove the slightly stronger inequality $-\max(\frac{\beta}{\sqrt{k_{t,i}}}, \frac{\epsilon(1-\gamma)}{m}) \cdot 2V_0 + \frac{1}{k_{t,i}} V_E \ge -\frac{\epsilon(1-\gamma)}{m}$. First note that if the second term dominates the max expression, then the inequality is automatically true, so we only have to deal with the situation when the first term dominates. In this case, the inequality takes the form $-\frac{\beta}{\sqrt{k_{t,i}}} \cdot 2V_0 + \frac{1}{k_{t,i}} V_E \ge -\frac{\epsilon(1-\gamma)}{m}$, which always holds because of our choice of R_E.

We show by induction that $V^{(t)}(\mathbf{x}) \ge V^\times(\mathbf{x}) - \Theta(\epsilon)$ and $Q^{(t)}(\mathbf{x},a) \ge Q^\times(\mathbf{x},a) - \Theta(\epsilon)$ for all $t = 0,1,2,\dots$ and all $(\mathbf{x},a) \in \mathbf{X} \times A$. The inequalities hold for $t=0$. When moving from step t to $t+1$,

$$\begin{aligned} Q^{(t+1)}(\mathbf{x},a) &= R(\mathbf{x},a) + \gamma \sum_{\mathbf{y}\in\mathbf{X}} \widehat{P}_t(\mathbf{y} \mid \mathbf{x},a) V^{(t)}(\mathbf{y}) \\ &\ge R(\mathbf{x},a) + \gamma \sum_{\mathbf{y}\in\mathbf{X}} \widehat{P}_t(\mathbf{y} \mid \mathbf{x},a)(V^\times(\mathbf{y}) - \Theta(\epsilon)) \\ &\ge Q^\times(\mathbf{x},a) - \gamma\Theta(\epsilon) - \Theta((1-\gamma)\epsilon) \end{aligned}$$

for all $(\mathbf{x},a)$, where we applied the induction assumption and eq. (11). Consequently, $\max_{a\in A} Q^{(t+1)}(\mathbf{x},a) \ge \max_{a\in A} Q^\times(\mathbf{x},a) - \Theta(\epsilon)$ for all $\mathbf{x}$. Note that according to our assumptions, all entries of H are nonnegative as well as the entries of $G = \mathcal{N}(H^T)$, so multiplication by rows of HG is a monotonous operator, furthermore, all rows sum to 1, yielding $\sum_{\mathbf{x}\in\mathbf{X}} [HG]_{\mathbf{y},\mathbf{x}} \max_{a\in A} Q^{(t+1)}(\mathbf{x},a) \ge \sum_{\mathbf{x}\in\mathbf{X}} [HG]_{\mathbf{y},\mathbf{x}} (\max_{a\in A} Q^\times(\mathbf{x},a) - \Theta(\epsilon))$, that is,

$V^{(t+1)}(\mathbf{x}) \ge V^\times(\mathbf{x}) - \Theta(\epsilon)$.

5.2.5. Proximity of value functions

The rest of the proof is standard, so we give here a very rough sketch only. We define a cutoff horizon $H := \Theta(\frac{R_E}{1-\gamma} \log \frac{1}{\epsilon(1-\gamma)})$ and an escape event A which happens at timestep t if the agent encounters an unknown transition in the next H steps. We will separate two cases depending on whether $\Pr(A)$ is smaller than $\frac{\epsilon(1-\gamma)}{R_E}$ or not. If the probability of escape is low, then we can show that $Q^{FOIM}(\mathbf{x}_t,a_t) \ge Q^\times(\mathbf{x}_t,a_t) - \Theta(\epsilon)$.

Otherwise, if $\Pr(A)$ is large, then an unknown state-action component is found with significant probability. However, this can happen only at most $mN_f|A|k_0$ times (because all components become known after k_0 visits), which is polynomial, so the second case can happen only a polynomial number of times.

Finally, we remind that the statements are true only with probability $1 - \Theta(\delta)$. To round off the proof, we note that we are free to choose the constant in the definition of R_E (as it is hidden in the $\Theta(\cdot)$ notation), so we set it in a way that $\Theta(\epsilon)$ and $\Theta(\delta)$ become at most ϵ and δ, respectively.

6. Discussion

FOIM is conceptually very simple: the exploration-exploitation dilemma is resolved *without any* explicit exploration, action selection is always greedy. The model update and model solution are also at least as simple as the alternatives found in the literature. Further, FOIM has some favorable theoretical properties. FOIM is the first example to an RL algorithm that has a polynomial per-step computational complexity in FMDPs. To achieve this, we had to relax the near-optimality of the FMDP planner. The particular planner we used, FVI, runs in polynomial time, it does reach a bounded error, and the looseness of the bound depends on the quality of basis functions. In almost all time steps, FOIM gets ϵ-close to the FVI value function with high probability (for any pre-specified ϵ). The number of timesteps when this does not happen is polynomial. From a practical point of view, calling an FMDP model-solver in each iteration could be prohibitive. However, the model and the value function usually change very little after a single model update, so we may initialize FVI with the previous value function, and a few iterations might be sufficient.

Acknowledgements

This work has been supported by the EC NEST Perceptual Consciousness: Explication and Testing grant under contract 043261. Opinions and errors in this manuscript are the authors responsibility, they do not necessarily reflect the opinions of the EC or other project members. The first author has been partially supported by the Fulbright Scholarship.

References

Boutilier, C., Dearden, R., & Goldszmidt, M. (1995). Exploiting structure in policy construction. *International Joint Conference on Artificial Intelligence* (pp. 1104–1111).

Boutilier, C., Dearden, R., & Goldszmidt, M. (2000). Stochastic dynamic programming with factored representations. *Artificial Intelligence*, *121*, 49–107.

Brafman, R. I., & Tennenholtz, M. (2001). R-MAX - a general polynomial time algorithm for near-optimal reinforcement learning. *International Joint Conference on Artificial Intelligence* (pp. 953–958).

Guestrin, C., Koller, D., Parr, R., & Venkataraman, S. (2002). Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, *19*, 399–468.

Kearns, M. J., & Koller, D. (1999). Efficient reinforcement learning in factored MDPs. *International Joint Conference on Artificial Intelligence* (pp. 740–747).

Kearns, M. J., & Singh, S. (1998). Near-optimal reinforcement learning in polynomial time. *International Conference on Machine Learning* (pp. 260–268).

Koller, D., & Parr, R. (2000). Policy iteration for factored MDPs. *Uncertainty in Artificial Intelligence* (pp. 326–334).

Liberatore, P. (2002). The size of MDP factored policies. *AAAI Conference on Artificial Intelligence* (pp. 267–272).

Strehl, A. L. (2007). Model-based reinforcement learning in factored-state MDPs. *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning* (pp. 103–110).

Strehl, A. L., Diuk, C., & Littman, M. L. (2007). Efficient structure learning in factored-state MDPs. *AAAI Conference on Artificial Intelligence* (pp. 645–650).

Strehl, A. L., & Littman, M. L. (2005). A theoretical analysis of model-based interval estimation. *International Conference on Machine Learning* (pp. 856–863).

Szita, I., & Lőrincz, A. (2008a). Factored value iteration. *Acta Cybernetica*, *18*, 615–635.

Szita, I., & Lőrincz, A. (2008b). The many faces of optimism: a unifying approach. *International Conference on Machine Learning* (pp. 1048–1055).

Szita, I., & Lőrincz, A. (2009). Optimistic initialization and greediness lead to polynomial time learning in factored MDPs – extended version. http://arxiv.org/abs/0904.3352.