Pseudo-MDPs and Factored Linear Action Models

Hengshuai Yao, Csaba Szepesvári, Bernardo Ávila Pires
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada, T6G2E8
{hengshua,szepesva,bpires}@ualberta.ca

Xinhua Zhang
Machine Learning Research Group
National ICT Australia
Sydney and Canberra, Australia
Email: Xinhua.Zhang@nicta.com.au

***Abstract*—In this paper we introduce the concept of pseudo-MDPs to develop abstractions. Pseudo-MDPs relax the requirement that the transition kernel has to be a probability kernel. We show that the new framework captures many existing abstractions. We also introduce the concept of factored linear action models; a special case. Again, the relation of factored linear action models and existing works are discussed. We use the general framework to develop a theory for bounding the suboptimality of policies derived from pseudo-MDPs. Specializing the framework, we recover existing results. We give a least-squares approach and a constrained optimization approach of learning the factored linear model as well as efficient computation methods. We demonstrate that the constrained optimization approach gives better performance than the least-squares approach with normalization.**

I. Introduction

In reinforcement learning an agent chooses actions in a sequential manner to maximize its long term reward while observing state transitions [1, 2]. In this paper we consider model-based reinforcement learning where a model of the Markovian environment is built first. When a model is built, the main questions are whether the model can be efficiently solved and whether the policy derived from the approximate model is useful. To answer these questions in a general form, we introduce the framework of pseudo-MDPs. Pseudo-MDPs relax the requirement that the transition kernel has to be a probability kernel. We show that the new framework captures many existing abstractions, such as those derived from state-aggregation, or even RKHS embeddings of MDPs [3]. We also introduce the concept of factored linear action models; which is a special case of pseudo-MDPs. Again, the relation of factored linear action models and existing works are discussed. We develop a general theory for bounding the suboptimality of policies derived from pseudo-MDPs. Specializing the framework, we recover existing results by [3] in the case of using kernel features. We propose a general approximate value iteration (AVI) algorithm that solves a pseudo-MDP. The advantage of this algorithm is that it has convergence guarantee with linear function approximation comparing to popular linear approximate policy iteration algorithms such as LSPI [4]. We propose two approaches of learning a factored action model, including a least-squares approach and a constrained optimization approach. We provide an efficient solution for the constrained optimization approach using gradient descent methods.

II. Background and Notation

In this section we provide the necessary background on MDPs. We define a finite-action MDP as a 4-tuple $\mathcal{M} = (\mathcal{X}, \mathcal{A}, (\mathcal{P}^a)_{a\in\mathcal{A}}, (f^a)_{a\in\mathcal{A}})$, where $\mathcal{X}$ is a set of measurable states,[1] $\mathcal{A}$ is a finite set of actions[2]; for each $a \in \mathcal{A}$ action and state $x \in \mathcal{X}$, the "kernel" $\mathcal{P}^a$ assigns a probability measure to x, which we denote by $\mathcal{P}^a(\cdot|x)$ and f^a is a real-valued function over $\mathcal{X}$. We will consider discounted MDPs only and we denote the *discount factor* by $\gamma \in [0, 1)$. An MDP gives rise to a sequential decision process, where at each stage an action has to be chosen based on the past observations, leading to a next observed state X' sampled from $\mathcal{P}^a(\cdot|X = x)$, where X is the current state and a is the action chosen. While transitioning to X', a reward of $f^a(X, X')$ is incurred, which is also observed. The goal is to find a way of choosing the actions so that the expected total discounted sum of rewards incurred is maximized no matter how the process is started. A standard result [5] is that this can be achieved by following some stationary Markov policy α: Here, $\alpha : \mathcal{X} \times \mathcal{A} \to [0, 1]$ and for each state $x \in \mathcal{X}$, $\alpha(x, \cdot)$ is distribution over $\mathcal{A}$. Following α means that the state is $X_t \in \mathcal{X}$ at time t then the next action is chosen from $\alpha(X_t, \cdot)$: $A_t \sim \alpha(X_t, \cdot)$. In what follows we will simply call stationary Markov policies a policy and denote their set by $\Pi_{\mathcal{M}}$. We will denote by $V^\alpha(x)$ the total expected discounted reward incurred while following α from state x; $V^\alpha(x) = \mathbb{E}\Big\{ \sum_{t=0}^{\infty} \gamma^t f^{A_t}(X_t) \Big| X_0 = x, X_{t+1} \sim \mathcal{P}^{A_t}(\cdot|X_t), A_t \sim \alpha(X_t, \cdot), t = 0, 1, 2, \dots \Big\}$. The optimal value of state x is $V^*(x) = \sup_\alpha V^\alpha(x)$, giving rise to the optimal value function $V^* : \mathcal{X} \to \mathbb{R}$. For these definition to make sense we need to make some further assumptions. First, for a measure μ over some measurable set W, introduce $L^1(\mu)$ to denote the space of μ-integrable real-valued functions with domain W. Further, for a kernel $\mathcal{P}^a$ let $L^1(\mathcal{P}^a) = \cap_{x\in\mathcal{X}} L^1(\mathcal{P}^a(\cdot|x))$. We also let $L^1(\mathcal{P}) = \cap_{a\in\mathcal{A}} L^1(\mathcal{P}^a) = \cap_{a\in\mathcal{A}, x\in\mathcal{X}} L^1(\mathcal{P}^a(\cdot|x))$. We require that for any $a \in \mathcal{A}$, $f^a \in L^1(\mathcal{P}^a)$ and further that for any measurable set $U \subset \mathcal{X}$, $a \in \mathcal{A}$, $\mathcal{P}^a(U|\cdot) \in L^1(\mathcal{P})$ (in particular, $x \mapsto \mathcal{P}^a(U|\cdot)$ must be measurable). These ensure that the expectations are well-defined. Note that $L^1(\mathcal{P}^a)$ and

[1] In particular, any finite set would do, in which case measurability becomes nonrestrictive. The generalization for measurable spaces (which include measurable subsets of Euclidean spaces) is pursued as it comes essentially for free and allows one to consider "large" spaces.

[2] This assumption could be lifted without much work.

$L^1(\mathcal{P})$ are vector-spaces. As is well known, the optimal value function V^* satisfies the so-called "Bellman optimality equations" $V^*(x) = \max_{a\in\mathcal{A}} f^a(x) + \gamma \int \mathcal{P}^a(dx'|x)V(x'), x \in \mathcal{X}$. Furthermore, only V^* is the solution to these simultaneous equations.

For a normed vector space $\mathcal{V} = (\mathcal{V}, \|\cdot\|)$, the (induced) norm of an operator $T : \mathcal{V} \to \mathcal{V}$ is defined by $\|T\| = \sup_{V\in\mathcal{V}, V\neq 0} \|TV\|/\|V\|$. An operator is called a contraction if $\|T\| < 1$. The difference of two operators $T, \hat{T} : \mathcal{V} \to \mathcal{V}$ is defined via $(T - \hat{T})V = TV - \hat{T}V$. The supremum norm $\|\cdot\|_\infty$ of a (real-valued) function f over some set W is defined by $\|f\|_\infty = \sup_{w\in W} |f(w)|$. We will denote by $\delta_{x_0}(dx)$ the Dirac measure concentrated on x_0: $\int f(x)\delta_{x_0}(dx) = f(x_0)$ for any measurable f.

III. The Pseudo-MDP Framework

We shall consider abstracting MDPs into what we call "pseudo-MDPs". Let $\mathcal{S}$ be a measurable space. Recall that a signed measure μ over $\mathcal{S}$ maps measurable subsets of $\mathcal{S}$ to reals and satisfies $\mu(\cup_i S_i) = \sum_i \mu(S_i)$ for any countable family $(S_i)_i$ of disjoint measurable sets of $\mathcal{S}$. We call the tuple $\mathcal{N} = (\mathcal{S}, \mathcal{A}, (\mathcal{Q}^a)_{a\in\mathcal{A}}, (g^a)_{a\in\mathcal{A}})$ a *pseudo-MDP* if $\mathcal{Q}^a$ maps elements of $\mathcal{S}$ to signed measures over $\mathcal{S}$ ($\mathcal{Q}^a(\cdot|s) \doteq \mathcal{Q}^a(s,\cdot)$ is a signed measure over $\mathcal{S}$) and $g^a : \mathcal{S} \to \mathbb{R}$ is a measurable function. As for MDPs, we assume that $g^a \in L^1(\mathcal{Q})$ and for any measurable $U \subset \mathcal{S}$ and action $a \in \mathcal{A}$, $\mathcal{Q}^a(U|\cdot) \in L^1(\mathcal{Q})$.

The difference between a pseudo- and a "real" MDP is that in a pseudo-MDP $\mathcal{Q}^a(\cdot|s)$ does not need to be a probability measure. This can be useful when constructing abstractions: dropping the requirement that the transition kernel must be a probability measure increases the power of pseudo-MDPs. The concepts of policies and value functions extend to pseudo-MDPs with almost no change except for defining the value function of a policy β of $\mathcal{N}$, we consider the signed measures $\mu_{s,\beta}$ induced by $(\mathcal{Q}^a)_a$ and β over the set $(\{s\} \times \mathcal{A}) \times (\mathcal{S} \times \mathcal{A})^{\mathbb{N}}$ of trajectories starting at some state $s \in \mathcal{S}$. Then the value function of β is v^β defined by $v^\beta(s) = \int \sum_{t=0}^\infty \gamma^t g^{a_t}(s_t) d\mu_{s,\beta}(s_0, a_0, s_1, a_1, \ldots)$. We assume that v^β is finite-valued for any policy β of $\mathcal{N}$.

The purpose of constructing pseudo-MDPs is to create abstractions that facilitate efficient computation. However, for an abstraction to be of any use, we need to be able to use it to come up with good (near-optimal) policies in the source MDP. Denoting the abstracted, or *source MDP* by $\mathcal{M} = (\mathcal{X}, \mathcal{A}, (\mathcal{P}^a)_{a\in\mathcal{A}}, (f^a)_{a\in\mathcal{A}})$, the connection will be provided by a measurable map $\phi : \mathcal{X} \to \mathcal{S}$, which must be chosen at time of choosing $\mathcal{N}$. In what follows we fix the mapping ϕ.

We let $\Pi_\mathcal{M}, \Pi_\mathcal{N}$ be the space of policies in the original MDP and the pseudo-MDP, respectively. The map ϕ can be used to pull any policy of the pseudo-MDP back to a policy of the source MDP:

Definition 1 (Pullback Policy)**.** *Let $\mathcal{N}$ be a ϕ-abstraction of $\mathcal{M}$. The* pullback of policy $\beta \in \Pi_\mathcal{N}$ *is the policy $\alpha \in \Pi_\mathcal{M}$ that satisfies $\alpha(x,a) = \beta(\phi(x), a)$. The map that assigns α to β will be denoted by L and we will call it the pullback map (thus, $L : \Pi_\mathcal{N} \to \Pi_\mathcal{M}$ and $L(\beta)(x,a) = \beta(\phi(x), a)$, for any $x \in \mathcal{X}$, $a \in \mathcal{A}$).*

The power of pseudo-MDPs is that it provides a common framework for many MDP-abstractions that were considered previously in the literature. Some examples are as follows:

Example 1 (Finite Models)**.** *Let $\mathcal{S}$ be a finite set, for $s \in \mathcal{S}$, $a \in \mathcal{A}$, $\mathcal{Q}^a(\cdot|s)$ be a distribution over $\mathcal{S}$, $g^a : \mathcal{S} \to \mathbb{R}$ be an arbitrary function.*

Example 2 (Linear Action Models)**.** *Assume that $\mathcal{S} = \mathbb{R}^d$ where measurability is meant in the Borel sense. For each $a \in \mathcal{A}$, let $F^a \in \mathbb{R}^{d\times d}$, $f^a \in \mathbb{R}^d$. Then, for each $s \in \mathcal{S}$, $a \in \mathcal{A}$, U Borel measurable, $\mathcal{Q}^a(U|s) = \mathbb{I}_{\{F^a s\in U\}}$ and $g^a(s) = (f^a)^\top s$.*

Example 3 (Factored Linear Action Models)**.** *Let $\mathcal{S} = \mathcal{X}$, $\psi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$, $\xi : \mathcal{B} \to \mathbb{R}^d$, where $\mathcal{B}$ is the collection of measurable sets of $\mathcal{X}$. Then, for $x \in \mathcal{X}$, $a \in \mathcal{A}$, $U \in \mathcal{B}$, $\mathcal{Q}^a(U|x) = \xi(U)^\top \psi(x,a)$, while g^a is arbitrary.*[3]

In the first two examples $(\mathcal{Q}^a)_a$ are probability kernels. Discrete models are typically obtained in a process known as *state aggregation* [6] in which case $\phi : \mathcal{X} \to \mathcal{S}$ is assumed to be surjective and is known as the state-aggregation function. Given ϕ, one further chooses for each $s \in \mathcal{S}$ a distribution μ_s supported on $\phi^{-1}(s) = \{x \in \mathcal{X} \,|\, \phi(x) = s\}$. Then, $\mathcal{Q}^a$ is given by $\mathcal{Q}^a(U|s) = \int \mu_s(dx)\mathcal{P}^a(dx'|x)\mathbb{I}_{\{\phi(x')\in U\}}$ and $g^a(s) = \int f^a(x)\mu_s(dx)$. Linear action models arise when the transition dynamics underlying each action is approximated via a linear model, in which case ϕ is known as the "feature-map" [7]. Note that one can represent any finite MDP with linear models: Given a finite model $\mathcal{N}$ with state space $\mathcal{S} = \{1, \ldots, d\}$, define $\tilde{\mathcal{S}}$, the state space of the linear model, as the simplex of $\mathbb{R}^d$, $(F^a)_{j,i} = \mathcal{Q}^a(j|i)$, $f^a_i = g^a(i)$, $1 \le i, j \le d$.

Motivating examples. Here we give a few motivating examples for pseudo-MDPs. Pseudo-MDPs increase the search space for solving the original MDP. Note that Pseudo-MDPs include MDPs, and so we just have to demonstrate the advantage of pseudo-MDPs that are not MDPs.

Ex. 1 (normalized model has a bad estimation of the optimal value function). The MDP is, $\mathcal{A} = \{a\}, \mathcal{S} = \{1, 2\}, \mathcal{P}^a = [0.01, 0.99; 0, 1], g^a = [1, 0]$. $V^*(1) \approx 0.01, V^*(2) = 0$. The discount factor is 0.9. The pseudo-MDP is different with $\mathcal{Q}^a = [0.01, 0; 0, 1]$; the state space and others are all the same. Then we have $\hat{V}^* = V^*$. If normalizing the model (i.e., normalizing each row of $\mathcal{Q}^a$ by its L-1 norm, which gives an MDP), we have, $\bar{\mathcal{Q}}^a = [1, 0; 0, 1]$, and $\hat{V}^*(1) = 10, \hat{V}^*(2) = 0$.

Ex. 2 (normalized model has a bad policy output). The MDP is, $\mathcal{A} = \{a_1, a_2\}, \mathcal{S} = \{1, 2\}, \mathcal{P}^{a_1} = [0.01, 0.99; 0, 1], \mathcal{P}^{a_2} = [0, 1; 0, 1]$, $g^{a_1}(1,1) = 100, g^{a_1}(1,2) = 0$, $g^{a_1}(2,2) = g^{a_2}(2,2) = 10, g^{a_2}(1,2) = 100$. The discount factor is 0.9. The optimal policy is $\alpha^*(1) = \alpha^*(2) = a_2$. A figure is shown in Figure 1.

[3] A natural restriction on g^a would be to assume $g^a(x) = (f^a)^\top \psi(x,a)$.

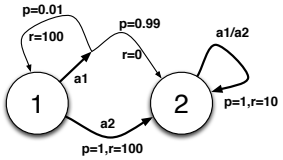


Fig. 1. A small MDP used in Ex. 2.

The pseudo-MDP parameters are, $\mathcal{Q}^{a_1} = [0.01, 0; 0, 1], \mathcal{Q}^{a_2} = [0, 1; 0, 1]$; all the other parameters are the same as the original MDP. The parameters of the normalized model (which is a valid MDP) are, $\bar{\mathcal{Q}}^{a_1} = [1, 0; 0, 1], \bar{\mathcal{Q}}^{a_2} = [0, 1; 0, 1]$; all the other parameters are the same as the original MDP. One can show that from the pseudo-MDP we can derive the optimal policy but the normalized model does not. In particular, the optimal policy according to the normalized model selects action a_1 at state 1.

Although linear action models are powerful, it may be difficult to compute a near-optimal policy in a linear action model. The idea of factored linear models is similar except that here the state space is unchanged; the "abstraction" happens because the transition kernel is written in a factored form: The map ψ extracts the features of state-action pairs, while the "features" of the sets one may arrive at are extracted by ξ. An interesting special case is when ξ takes the form

$$\xi(U) = \int_U f(x')\mu(dx'), \tag{1}$$

where μ is a signed measure over $\mathcal{X}$ and $f : \mathcal{X} \to \mathbb{R}^d$ is measurable. When μ is a counting measure with finite support $\mathcal{X}' \subset \mathcal{X}$, we have $\xi(U) = \sum_{x' \in \mathcal{X}' \cap U} f(x')$ and $\mathcal{Q}^a(U|x) = \sum_{x' \in \mathcal{X}'} f(x')^\top \psi(x, a)$. In this case, under some additional conditions the optimal policy can be computed efficiently. Indeed, if $\hat{V}^*$ denotes the optimal value function for the factored model, from Bellman's optimality equation,

$$\begin{aligned}\hat{V}^*(x) &= \max_{a \in \mathcal{A}} g^a(x) + \gamma \left(\sum_{x' \in \mathcal{X}'} \hat{V}^*(x') f(x') \right)^\top \psi(x, a) \\ &= \max_{a \in \mathcal{A}} (\hat{T}^a \hat{V}^*)(x),\end{aligned}$$

where the last equation defines the operators $\hat{T}^a$. By this equation, knowing $\hat{V}^*$ at states in $\mathcal{X}'$ suffices to compute an optimal action of $\mathcal{N}$ at any state $x \in \mathcal{X}$. The Bellman optimality equation will be guaranteed to have a solution if $\hat{T}^a$ is a contraction in the $\|\cdot\|_\infty$-norm, which holds if $|\sum_{x' \in \mathcal{X}'} f(x')^\top \psi(x, a)| \le 1$ for any $(x, a) \in \mathcal{X} \times \mathcal{A}$. Using Bellman's optimality equation again, we see that $\hat{V}^*|_{X'}$ is the optimal value function of the *finite* pseudo-MDP

$$(\mathcal{X}', (\mathcal{Q}^a|_{\mathcal{X}' \times 2^{\mathcal{X}'}})_{a \in \mathcal{A}}, (g^a|_{\mathcal{X}'})_{a \in \mathcal{A}}) \tag{2}$$

and, as such, it can be found, e.g., by any dynamic programming algorithm.

Given a finite model $\mathcal{N} = (\mathcal{S}, \mathcal{A}, (\mathcal{Q}^a)_{a \in \mathcal{A}}, (g^a)_{a \in \mathcal{A}})$ with $\mathcal{S} = \{1, \dots, d\}$ and a surjective map $\phi : \mathcal{X} \to \mathcal{S}$, pick $\mu(\cdot|i)$ so that for each $i \in \mathcal{S}$, $\mu(\cdot|i)$ is a probability distribution supported on $\phi^{-1}(i) = \{x \in \mathcal{X} \,|\, \phi(x) = i\}$. Define the probability kernels $(\hat{\mathcal{P}}^a)_{a \in \mathcal{A}}$ by $\hat{\mathcal{P}}^a(dx'|x) = \sum_{j \in \mathcal{S}} \mu(dx'|j) \mathcal{Q}^a(j|\phi(x))$. By choosing $\xi_i(dx) = \mu(dx|i)$, $\psi_i(x, a) = \mathcal{Q}^a(i|\phi(x))$, $1 \le i \le d$, we see that $\hat{\mathcal{P}}^a(U|x) = \xi(U)^\top \psi(x, a)$, thus a finite model gives rise to a factored model.

Now, consider the following construction: Let $\mathcal{Z} = \mathcal{X} \times \mathcal{A}$, $(z_1, x'_1), \dots, (z_n, x'_n) \in \mathcal{Z} \times \mathcal{X}$, $\nu(\cdot|x)$ a probability kernel over $\mathcal{X}$, and $k : \mathcal{Z} \times \mathcal{Z} \to [0, \infty)$ a function such that $\sum_{j=1}^n k((x, a), z_i) > 0$ for any $(x, a) \in \mathcal{Z}$. Define $\xi_i(dx') = \nu(dx'|x'_i)$ and $\psi_i(x, a) = k((x, a), z_i) / \sum_{j=1}^n k((x, a), z_j)$. It is not hard to see that the resulting factored model is a generalization of the *kernel-based model* of [8] who chooses $\nu(dx'|x'_i) = \delta_{x'_i}(dx')$. In this case, ξ can be put in the form (1) with $f_i(x') = \frac{\mathbb{I}_{\{x' = x'_i\}}}{|\{j \,|\, x'_j = x'_i\}|}$ and $\mu(dx') = \sum_{j=1}^n \delta_{x'_j}(dx')$, thus the model can be "solved" efficiently. The model of [3] that embeds the transition kernels into reproducing kernel Hilbert spaces can also be seen to be a factored linear model, though to allow this we need to replace the range of ξ and ψ with a Hilbert space. Details are left out due to the lack of space.

A. A Generic Error Bound

The main purpose of this section is to derive a bound on how well the pullback of a near-optimal policy of a pseudo-MDP will do in the source MDP. Before stating this result, we need some definitions. Given any measurable function v over $\mathcal{S}$, we let V_v denote the function over $\mathcal{X}$ defined by $V_v(x) = v(\phi(x))$: V_v is called the pullback of v. We also introduce a left inverse $l : \Pi_{\mathcal{M}} \to \Pi_{\mathcal{N}}$ to L, which we call a *pushforward map*. Thus, $l(L(\beta)) = \beta$ holds for any $\beta \in \Pi_{\mathcal{N}}$. Note that to ensure that L has a left inverse, ϕ must be surjective:

Assumption A1 ϕ is surjective.

When ϕ is surjective, it is easy to see that a left inverse of L indeed exists and, in fact, there could be multiple left inverses. The pushforward map is a theoretical construction in the sense that it is only used in characterizing the "power" of abstractions (it is not used algorithmically). This allows one to choose the best pushforward map that gives the tightest error bounds.

A pushforward and a feature map together give rise to the concept of approximate value functions:

Definition 2 (Approximate Value Function)**.** *Fix a pushforward map l and a feature map ϕ. Given a policy $\alpha \in \Pi_{\mathcal{M}}$, we call $v^{l(\alpha)}$ the value-function of α under l in $\mathcal{N}$. Further, we let $V_v^\alpha \doteq V_{v^{l(\alpha)}}$ be the $\mathcal{N}$-*induced approximate value function *underlying policy α.*[4]

Let $\mathcal{B}(\mathcal{X}) = (\mathcal{B}(\mathcal{X}), \|\cdot\|)$ be a normed subspace of $L^1(\mathcal{P})$: $\mathcal{B}(\mathcal{X}) = \{V : \mathcal{X} \to \mathbb{R} \,|\, V \in L^1(\mathcal{P}), \|V\| < \infty\}$. We use the norm $\|\cdot\|$ associated with $\mathcal{B}(\mathcal{X})$ to measure the magnitude of the errors introduced by $\mathcal{N}$: We call

$$\epsilon(\alpha) = \|V^\alpha - V_v^\alpha\| \tag{3}$$

[4]In fact, in addition to $\mathcal{N}$, both l and ϕ influence V_v^α.

the evaluation error of policy α induced by $\mathcal{N}$.

To compare policies we will use the expected total discounted reward where the initial state is selected from some fixed distribution, which we will denote by ρ. Given any $V : \mathcal{X} \to \mathbb{R}$, define $V_\rho = \int_{x\in\mathcal{X}} V(x)\rho(dx)$. Then $V_\rho^\alpha = \int_{x\in\mathcal{X}} V^\alpha(x)\rho(dx)$ gives the expected total discounted reward collected while following α assuming that the initial state is selected from ρ. Further, for a function $V \in L^1(\mathcal{P})$, define its $L^1(\rho)$-norm by $\|V\|_{L^1(\rho)} = \int |V(x)|\rho(dx)$ and let $K_\rho = \sup_{V\in\mathcal{B}(\mathcal{X})} \|V\|_{L^1(\rho)}/\|V\|$. We will denote by $\phi_*(\rho)$ the *pushforward of ρ under ϕ*: $\phi_*(\rho)$ is a probability measure on $\mathcal{S}$: it is the distribution of $\phi(X)$ where $X \sim \rho$.

With this, we can present our first main result which bounds the suboptimality of the pullback of the pseudo-MDP's optimal policy:[5]

Theorem 1. *Let $\alpha^* \in \arg\max_{\alpha\in\Pi_\mathcal{M}} V_\rho^\alpha$, $\beta^* \in \arg\max_{\beta\in\Pi_\mathcal{N}} v^\beta_{\phi_*(\rho)}$ and let $\alpha_L^* = L(\beta^*)$ be the pullback of β^*. Then, under A1,*

$$V_\rho^{\alpha^*} - K_\rho(\epsilon(\alpha^*) + \epsilon(\alpha_L^*)) \le V_\rho^{\alpha_L^*} \le V_\rho^{\alpha^*}.$$

The theorem shows that the quality of the policy derived from an optimal policy of the pseudo-MDP is governed by the error induced by $\mathcal{N}$ on the value functions of policies α^*, α_L^* alone. Thus, it suggests that when considering the construction of $\mathcal{N}$, one should concentrate on the evaluation error of these two policies. *The result is remarkable because it suggests that the common objection against model learning according to which model learning is hard because a good model has to capture all the details of the world might not be as well founded as one may think it is.* Of course, the difficulty is that while β^* may be accessible (given $\mathcal{N}$), α^* is hardly available. Nevertheless, the result suggests an iterative approach towards constructing $\mathcal{N}$, which we will explore later.

The policy evaluation error defined in (3) depends on the norm chosen for the functions over $\mathcal{X}$. If one chooses the supremum norm, Theorem 1 immediately gives the following result:

Corollary 2. *Let $\|\cdot\| = \|\cdot\|_\infty$ in* (3)*. Then, under A1, for any optimal policy α^* of $\mathcal{M}$ and optimal policy β^* of $\mathcal{N}$, $\|V^{\alpha_L^*} - V^*\|_\infty \le \epsilon(\alpha^*) + \epsilon(\alpha_L^*)$, where $\alpha_L^* = L(\beta^*)$.*

Note that the definition of α^* and β^* in Theorem 1 is different from the definition used in this corollary. While here α^*, β^* are required to be optimal, in Theorem 1 they are optimal only in a weaker, average sense. Note that choosing the norm in (3) to be the supremum norm makes $K_\rho = 1$ for any distribution ρ (which is favourable), but can increase the values of $\epsilon(\alpha^*)$ and $\epsilon(\alpha_L^*)$. Hence, the norm that optimizes the bound may very well be different from the supremum norm.

B. Injective Feature Maps

When the feature map $\phi : \mathcal{X} \to \mathcal{S}$ is injective (and thus invertible), the generic bound of the previous section gives rise to a bound of a particularly appealing form. *When ϕ is a bijection, we can identify $\mathcal{S}$ with $\mathcal{X}$ without loss of generality and choose ϕ to be the identity map, an assumption that we will indeed make in this section*. The factored linear action model considered in the previous section gives a useful example when $\mathcal{S} = \mathcal{X}$. In general, when $\mathcal{S} = \mathcal{X}$, the approximation happens through "compressing" the transition kernel.

For simplicity, we also assume that $g^a \equiv f^a$, i.e., the rewards are not approximated (the extension of the results to the general case is trivial). In summary, the pseudo-MDP considered in this section takes the form $\mathcal{N} = (\mathcal{X}, \mathcal{A}, (\hat{\mathcal{P}}^a)_{a\in\mathcal{A}}, (f^a)_{a\in\mathcal{A}})$ (we replace $\mathcal{Q}^a$ by $\hat{\mathcal{P}}^a$ to emphasize that the approximate kernels are now over the state space of the source MDP).

When $\max_{a\in\mathcal{A}} \|\hat{\mathcal{P}}^a\|_1 \le 1$, Corollary 2 together with standard contraction arguments leads to the following result:

Theorem 3. *Let $\mathcal{N} = (\mathcal{X}, \mathcal{A}, (\hat{\mathcal{P}}^a)_{a\in\mathcal{A}}, (f^a)_{a\in\mathcal{A}})$ be a pseudo-MDP such that $\max_{a\in\mathcal{A}} \|\hat{\mathcal{P}}^a\|_1 \le 1$. Then, for any optimal policy $\hat{\alpha}^*$ of $\mathcal{N}$,*

$$\|V^{\hat{\alpha}^*} - V^*\|_\infty \le \tfrac{2\gamma}{(1-\gamma)^2} \min\{ \|(\hat{\mathcal{P}} - \mathcal{P})V^*\|_\infty, \|(\hat{\mathcal{P}} - \mathcal{P})v^*\|_\infty \}.$$

Again, we see that it suffices if $\hat{\mathcal{P}}$ is a good approximation to $\mathcal{P}$ at V^*. Since V^* is unknown, in practice one may choose a normed vector-space $\mathcal{F}$ of functions over $\mathcal{X}$ and construct $\hat{\mathcal{P}}$ such that it is a good approximation to $\mathcal{P}$ over $\mathcal{F}$ in the sense that $\epsilon(\mathcal{F}) = \sup_{V\in\mathcal{F}, \|V\|_\mathcal{F}=1} \|(\hat{\mathcal{P}} - \mathcal{P})V\|_\infty$ is small (here, $\|\cdot\|_\mathcal{F}$ denotes the norm that comes with $\mathcal{F}$). Can this approach succeed? Let $\Delta\mathcal{P} = \hat{\mathcal{P}} - \mathcal{P}$. Then, for any $V \in \mathcal{F}$, $\|\Delta\mathcal{P}V^*\|_\infty \le \|(\hat{\mathcal{P}} - \mathcal{P})(V^* - V)\|_\infty + \|\Delta\mathcal{P}V\|_\infty \le 2\|V^* - V\|_\infty + \epsilon(\mathcal{F})\|V\|_\mathcal{F}$. Taking the infimum over $V \in \mathcal{F}$, we get the following result:

Corollary 4. *Under the same conditions as in Theorem 3, for any optimal policy $\hat{\alpha}^*$ of $\mathcal{N}$, $\|V^{\hat{\alpha}^*} - V^*\|_\infty \le \frac{2\gamma}{(1-\gamma)^2} \inf_{V\in\mathcal{F}} 2\|V^* - V\|_\infty + \epsilon(\mathcal{F})\|V\|_\mathcal{F}$.*

Thus, the approach will be successful as long as our bet that V^* is close to $\mathcal{F}$ is correct and in particular if the L^∞-projection of V^* to $\mathcal{F}$ has a small $\mathcal{F}$-norm. Note that Corollary 4 can be viewed as a generalization/specialization of Theorem 3.2 of [3].[6] The assumption $\max_{a\in\mathcal{A}} \|\hat{P}^a\|_1 \le 1$ is relatively mild. In the next section, we show how to learn models such that this assumption is satisfied.

IV. Learning Factored Linear Models

In this section we propose two approaches of learning factored linear models including a least-squares approach and a constrained optimization approach. We then give a procedure of solving the resulting pseudo-MDPs.

[5] All the proofs of the theoretical results in this paper is available in an extended version.

[6] The specialization comes from the fact that while Theorem 3.2 considers all kinds of approximations, we concentrate on the approximation induced by the approximate model as we find the approach that separates this from other approximation terms much cleaner.

A. Least-Squares Approach

In this section we show how factored linear models arise from a least-squares approach, essentially reproducing the model of [3] in a finite-dimensional setting from simple first principles (thus, hopefully catching the interest of readers who would shy away from the infinite dimensional setting considered by [3]). The factored linear model that arises will be the basis of the feature iteration method proposed in the next section.

As before, we will denote $\mathcal{Z} = \mathcal{X} \times \mathcal{A}$. Choose $V \in L^1(\mathcal{P})$ and suppose that we are interested in estimating the function $(x,a) \mapsto \int \mathcal{P}^a(dx'|x)V(x')$ where $(x,a) \in \mathcal{Z}$. Let $Z = (X,A)$ be a random state-action pair sampled from a distribution with full support over $\mathcal{Z}$ and $X' \sim P^A(\cdot|X)$. Then, $\int \mathcal{P}^a(dx'|x)V(x') = \mathbb{E}[V(X')|Z = (x,a)]$. Assume that we are given a mapping $\psi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$ to extract features based on state-action pairs and our goal is to find the best linear estimator $z \mapsto u^\top \psi(z)$ based on ψ of the function $z \mapsto \mathbb{E}[V(X')|Z = z]$. The parameter vector of the estimator that minimizes the expected squared error is $u^*(V) \in \arg\min_{u \in \mathbb{R}^d} \mathbb{E}[\big(V(X') - u^\top \psi(Z)\big)^2]$. A simple calculation shows that $u^*(V) = \mathbb{E}[\psi(Z)\psi(Z)^\top]^\dagger \mathbb{E}[\psi(Z)V(X')]$, where $M^\dagger$ denotes the pseudo-inverse of matrix M.

In practice, $u^*(V)$ is approximated based on a finite dataset, $(\langle z_i, x_i' \rangle, i = 1, \ldots, n)$. Defining $u_n(V) = \left(\Psi^\top \Psi\right)^\dagger \Psi^\top \bar{V}$, where $\Psi = [\psi(z_i)^\top] \in \mathbb{R}^{n \times d}$ and $\bar{V}_i = V(x_i')$, $1 \le i \le n$, $u_n(V)$ optimizes the squared prediction error of $u^\top \psi(z)$ computed over $(\langle z_i, x_i' \rangle, i = 1, \ldots, n)$. Introducing $F = \Psi \left(\Psi^\top \Psi\right)^\dagger$ and letting $F_{i:}$ denote the ith row of F (i.e., $F_{i:} \in \mathbb{R}^{1 \times d}$), we calculate

$$\begin{aligned} u_n(V)^\top \psi(x,a) &= \bar{V}^\top F \psi(x,a) \\ &= \int \sum_{i=1}^n V(x') \delta_{x_i'}(dx') F_{i:} \psi(x,a)\,. \end{aligned} \tag{4}$$

Thus with $\xi(dx') = \sum_{i=1}^n \delta_{x_i'}(dx') F_{i:}^\top$, if $\hat{\mathcal{P}}^a(dx'|x) = \xi(dx')^\top \psi(x,a)$ then given ψ, $(x,a) \mapsto \int \hat{\mathcal{P}}^a(dx'|x)V(x')$ is the best linear least-squares estimator of $(x,a) \mapsto \int \mathcal{P}^a(dx'|x)V(x')$ *for any* $V \in L^1(\mathcal{P})$. In this sense, $(\hat{\mathcal{P}}^a)_{a \in \mathcal{A}}$ is the "best" estimate of $(\mathcal{P}^a)_{a \in \mathcal{A}}$.

Since $(\hat{\mathcal{P}}^a)_{a \in \mathcal{A}}$ is of the form (1) with $f(x') = \mathbb{I}_{\{x' = x_i'\}} F_{i:}^\top$,[7] the discussion after (1) applies: The approximate model can be solved with finite resources up to any desired accuracy.

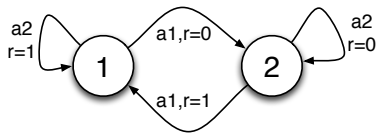
For computational purposes, it is worthwhile to define $\pi : \mathcal{Z} \to \mathbb{R}^n$ using $\pi_i(x,a) = F_{i:}\psi(x,a)$. Then, the prediction of $\mathbb{E}[V(X')|Z = (x,a)]$ simply becomes[8]

$$u_n(V)^\top \psi(x,a) = \bar{V}^\top \pi(x,a).$$

[7] Strictly speaking this holds when no data point is repeated. The best way to address problems with duplicated datapoints would be to allow $\xi(dx') = \sum_{i=1}^n \delta_{x_i'}(dx') f_i(x')$, which does not change the discussion after (1).

[8] When using kernels to generate the features, the matrix Ψ will be an $n \times n$ symmetric matrix and the formula given here reduces to that of [3].

Fig. 2. An MDP example used to show that least-squares model does not guarantee the L1-norm constraint.



As considered beforehand, if

$$\|\pi(x,a)\|_1 \le 1, \qquad x \in \{x_1', \ldots, x_n'\},\, a \in \mathcal{A}\,, \tag{5}$$

holds, the Bellman optimality operator of the finite pseudo-MDP given by (2) underlying $(\hat{\mathcal{P}}^a)_a$ will be a contraction and thus $\hat{V}^*$, the optimal value function in the pseudo-MDP will exist.

The following counterexample shows that (5) is not guaranteed to hold. Consider an MDP with $\mathcal{S} = \{1,2\}$, and $\mathcal{A} = \{1,2\}$. The (state) feature vectors are, $\phi(1) = 1, \phi(2) = 2$. Let $D^{a_j} = diag(d_{1,j}, d_{2,j})$ with $d_{i,j}$ being the frequncy of taking a_j at state i, $i = 1,2; j = 1,2$. Let the samples be arranged such that samples of action a_1 appear first. Let $\Phi^\top = [1,2]$. We have

$$\begin{aligned} (\Psi^\top \Psi)^\dagger &= \begin{pmatrix} (\Phi^\top D^{a_1} \Phi)^\dagger & 0 \\ 0 & (\Phi^\top D^{a_2} \Phi)^\dagger \end{pmatrix} \\ &= \begin{pmatrix} 1/(d_{1,1} + 4d_{2,1}) & 0 \\ 0 & 1/(d_{1,2} + 4d_{2,2}) \end{pmatrix} \end{aligned}$$

Now

$$\begin{aligned} \|\pi(1,a_1)\|_1 &= \|\Psi(\Psi^\top \Psi)^\dagger \psi(1,a_1)\|_1 \\ &= \sum_{i=1}^n \psi(x_i,b_i)^\top [1/(d_{1,1} + 4d_{2,1}), 0]^\top \\ &= (d_{1,1} + 2d_{2,1})/(d_{1,1} + 4d_{2,1}). \end{aligned}$$

Set $d_{1,1} = 9, d_{2,1} = 1$, we have $\|\pi(1,a_1)\|_1 \approx 0.8462$. Set $d_{1,2} = 1, d_{2,2} = 9$. Similarly, we have $\|\pi(2,a_1)\|_1 = 2\|\pi(1,a_1)\|_1 \approx 1.6923$, $\|\pi(1,a_2)\|_1 = 0.5135$, and $\|\pi(2,a_2)\|_1 = 1.0270$.

Now look at the MDP in Figure 2, with $\mathcal{P}^{a_1} = [0,1;1,0]; \mathcal{P}^{a_2} = [1,0;0,1]$. $g^{a_2}(1,1) = g^{a_1}(2,1) = 1.0, g^{a_1}(1,2) = 0.0, g^{a_2}(2,2) = 0$. The discount factor is 0.9. The features are specified as above. We used 9 pairs of $(x = 1, a = 1)$, one pair of $(x = 2, a = 1)$; one pair of $(x = 1, a = 2)$ and 9 pairs of $(x = 2, a = 2)$. Note this guarantees the same model as above. The L1-norm constraint is not satisfied. The AVI procedure using the model quickly diverges if using the iterative procedure for policy evaluation; the procedure has policy oscillation if using a direct solver.

One solution is to normalize each $\pi(x,a)$ by the L-1 norm [3]. In the next section, we propose another solution.

B. The Constraint Approach

We propose to modify the least-squares fitting problem by adding constraint (5). The resulting least squares problem can

be formulated in terms of the matrix $F \in \mathbb{R}^{n\times d}$:

$$\begin{aligned} &\text{minimize } \|\Psi F^\top - I_{n\times n}\|_F^2 \qquad (6)\\ &\text{subject to } \sum_{j=1}^{n} |F_{j:}\psi(x'_i,a)| \le 1, \qquad a\in\mathcal{A}, 1\le i\le n, \end{aligned}$$

where $I_{n\times n}$ is the $n\times n$ identity matrix and $\|\cdot\|_F$ denotes the Frobenius norm. Note that the objective function is a convex quadratic function, while the constraints can be rewritten as linear constraints. To explain the objective function, note that by (4), for $V \in L^1(\mathcal{P})$ arbitrary, the least-squares prediction of $\int \mathcal{P}^{a_i}(dx'|x_i)V(x') \approx V(x'_i)$ is $\bar{V}^\top F\psi(x_i,a_i)$. Hence, F should be such that $\sum_{i=1}^n (\bar{V}^\top F\psi(x_i,a_i) - V(x'_i))^2 = \|(\Psi F^\top - I_{n\times n})\bar{V}\|_2^2$ is small.

Choosing $V \in \{e_1,\dots,e_n\}$ and summing, we get the objective of (6). Note that this suggests alternative objectives, such as $\sup_{\bar{V}:\|\bar{V}\|_2\le 1} \|(\Psi F^\top - I_{n\times n})\|_2 = \|\Psi F^\top - I\|_2$, which is again convex.

Let $f = (F_{1:},\dots,F_{n:})^\top \in \mathbb{R}^{nd}$, $e = (e_1^\top,\dots,e_n^\top)^\top$. The objective function of (6) can be written as

$$\|\Psi F^\top - I_{n\times n}\|_F^2 = \sum_{i=1}^n \|\Psi F_{i:}^\top - e_i\|_2^2 = \|Hf - e\|_2^2, \quad (7)$$

where $H \in \mathbb{R}^{n^2\times nd}$ is defined by

$$H = \begin{pmatrix} \Psi & 0 & \dots & 0 \\ 0 & \Psi & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \Psi \end{pmatrix}.$$

Note that $H^\top H \in \mathbb{R}^{nd\times nd}$ is given by

$$H^\top H = \begin{pmatrix} \Psi^\top\Psi & 0 & \dots & 0 \\ 0 & \Psi^\top\Psi & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \Psi^\top\Psi \end{pmatrix}$$

To put (6) into the canonical form of linearly constrained quadratic optimization, introduce the variables $\xi_{j,ia} = |F_{j:}\psi(x'_i,a)|$. Further, let $S_j \in \mathbb{R}^{d\times nd}$ be the block matrix $S_j = (0,\dots,0,I_{d\times d},0,\dots,0)$ so that $S_j f = F_{j:}^\top$. With this, we can write (6) as

$$\begin{aligned} &\text{minimize } f^\top H^\top H f - 2e^\top H f \\ &\text{subject to} \\ &\xi_{j,ia} \ge \psi(x'_i,a)^\top S_j f, \quad 1\le i,j\le n, a\in\mathcal{A}, \\ &\xi_{j,ia} \ge -\psi(x'_i,a)^\top S_j f, \quad 1\le i,j\le n, a\in\mathcal{A}, \\ &\sum_{j=1}^n \xi_{j,ia} \le 1, \quad 1\le i\le n, a\in\mathcal{A}. \end{aligned}$$

Denote the transition kernels derived from the solution of (6) by $(\tilde{\mathcal{P}}^a)_{a\in\mathcal{A}}$ and the resulting pseudo-MDP by $\tilde{N}$.

To summarize, to learn a model and to use it to produce a policy, the following steps are followed: *(i)* data is collected of the form $(\langle z_i, r_i, x'_i\rangle, i=1,\dots,n)$, where $z_i = (x_i,a_i) \in \mathcal{Z}$, $x'_i \in \mathcal{X}$ and $r_i \in \mathbb{R}$ (the intention is that $\langle z_i, r_i, x'_i\rangle$ represents a transition sampled from the true model); *(ii)* based on the data, matrix F and then the normalized table $(\tilde{\pi}_j(x_i,a))_{1\le i,j\le n, a\in\mathcal{A}}$ are calculated; *(iii)* value- or policy-iteration [9] is used to find the optimal value function of the finite pseudo-MDP with n states where the reward at state i is r_i, and the transition kernel is $\mathcal{Q}^a(j|i) = \tilde{\pi}_j(x_i,a_i)$. Denote the computed optimal value function by v. We will view v as an n-dimensional vector over x'_i. Finally, an optimal action at state $x \in \mathcal{X}$ of underlying the model that uses $(\tilde{\mathcal{P}}_a)_{a\in\mathcal{A}}$ is obtained by computing $\mathrm{argmax}_{a\in\mathcal{A}}\, g^a(x) + \gamma v^\top \tilde{\pi}(x,a)$. The pseudo code of the proposed AVI algorithm is shown in Algorithm 1.

Algorithm 1 A generalized AVI algorithm that is based on an approximate model extending kernel embedding [3].

Input: A set of samples $(\langle x_i,a_i,r_i,x'_i\rangle)_{i=1,2,\dots,n}$, a feature mapping $\psi : \mathcal{X}\times\mathcal{A} \to \mathbb{R}^d$, and a matrix $F \in \mathbb{R}^{n\times d}$.
/*Those x'_i that are terminal have $\psi(x'_i,a) = \mathbf{0}$, which ensures $\pi(x'_i,a) = \mathbf{0}$; for $\forall a \in \mathcal{A}$. */
Output: A policy α and an estimate of V^α at the samples $(x'_i)_{i=1,2,\dots,n}$
..... **Modeling** .....
Compute the action model π for all x'_i:
For $i = 1,\dots,n$
  For each action a
    Compute $\pi(x'_i,a) = F\psi(x'_i,a)$
  End
End
..... **AVI** .....
Initialize policy α /*α is specified at $(x'_i)_{i=1,2,\dots,n}$ */
Initialize $\bar{V}^\alpha$ /*n-dimensional vector; $\bar{V}^\alpha(i)$ estimates $V^\alpha(x'_i)$ */
Generate vector $\bar{r}$ with $\bar{r}(i) = r_i$.
Repeat
  Solve $\bar{V}^\alpha$ for the current policy: /* iteratively or directly*/
    $\bar{V}^\alpha(i) = \pi(x'_i,\alpha(x'_i))^\top(\bar{r} + \gamma\bar{V}^\alpha), \quad i = 1,2,\dots,n,$
  For $i = 1,\dots,n$
    Compute $\alpha(x'_i) = \arg\max_{a\in\mathcal{A}} \pi(x'_i,a)^\top(\bar{r} + \gamma\bar{V}^\pi)$.
  End
End

We can prove that when the constraints in the optimization problem in equation (6) are removed, the resulting solution is equal to the least-square solution.

We need an efficient solver for the constraint approach for which off-the-shelf softwares are very slow. Let matrix $A \in \mathbb{R}^{d\times|\mathcal{A}|n}$ be $A = [\Psi_{a_1}^\top, \Psi_{a_2}^\top, \dots, \Psi_{a_{|\mathcal{A}|}}^\top]$ where Ψ_{a_k} is in $\mathbb{R}^{n\times d}$ and $\Psi_{a_k}(i,j) = \psi_j(x_i,a_k)$. The optimization problem can be written as

$$\begin{aligned} &\min_{F:\|A^\top F^\top\|_{1,\infty}\le 1} \frac{1}{2}\|F\Psi^\top - I\|_F^2 \\ &\Leftrightarrow \min_{F,Y:Y=A^\top F^\top} \frac{1}{2}\|F\Psi^\top - I\|_F^2 + \delta(\|Y\|_{1,\infty} \le 1). \end{aligned}$$

where $\delta(\cdot) = 0$ if $\cdot$ is true and ∞ otherwise. $\|Z\|_{p,q} := (\sum_i(\sum_j |Z_{ij}|^p)^{q/p})^{1/q}$, i.e., the ℓ_q norm of $(y_1,y_2,\dots)^\top$ where y_i is the ℓ_p norm of the i-th row of Z. It is well known that the dual norm of ℓ_p norm is the ℓ_{p^*} norm, where $1/p + 1/p^* = 1$. The dual norm of $\|\cdot\|_{p,q}$ is $\|\cdot\|_{p^*,q^*}$.

Note that we are deliberately decoupling $A^\top F^\top$ and Y. We solve this problem by applying Alternating Direction Method of Multipliers (ADMM) [10], which gradually enforces $Y =$

$A^\top F^\top$ through the minimization of augmented Lagrangian

$$L(F, Y, \Lambda) = \frac{1}{2}\|F\Psi^\top - I\|_F^2 + \delta(\|Y\|_{1,\infty} \le 1) - \text{tr}(\Lambda^\top(Y - A^\top F^\top)) + \frac{1}{2\mu}\|Y - A^\top F^\top\|_F^2,$$

in the following steps:

1. Initialize F_0 and set $Y_0 = A^\top F_0^\top$ and $\Lambda_0 = \mathbf{0}$. $t \leftarrow 1$.
2. $Y_t \leftarrow \arg\min_Y L(F_{t-1}, Y, \Lambda_{t-1})$.
3. $F_t \leftarrow \arg\min_F L(F, Y_t, \Lambda_{t-1})$.
4. $\Lambda_t \leftarrow \Lambda_{t-1} + \frac{1}{\mu}(A^\top F_t^\top - Y_t)$.
5. $t \leftarrow t + 1$, and go to step 2. Terminate if the difference between Y_t and $A^\top F_t^\top$ falls below some threshold.

Step 2 essentially solves

$$\min_{Y:\|Y\|_{1,\infty}\le 1} \frac{1}{2}\|Y - Z_t\|_F^2,$$

where $Z_t = \mu\Lambda_{t-1} + A^\top F_{t-1}^\top$. Note the constraint and objective are decoupled along rows, and therefore it suffices to solve

$$\min_{\mathbf{y}:\|\mathbf{y}\|_1\le 1} \frac{1}{2}\|\mathbf{y}^\top - (Z_t)_{i:}\|_F^2,$$

where $(Z_t)_{i:}$ stands for the i-th row of Z_t. This can be solved in linear time by, e.g., [11].

Step 3 minimizes an unconstrained quadratic function in F:

$$\min_F \frac{1}{2}\text{tr}(F\Psi^\top\Psi F^\top) + \frac{1}{2\mu}\text{tr}(FAA^\top F^\top) - \text{tr}(C_t^\top F),$$

where $C_t = \Psi - \Lambda_{t-1}^\top A^\top + \frac{1}{\mu}Y_t^\top A^\top$ and C_t changes over iteration. Setting the gradient to $\mathbf{0}$, a solution is optimal if and only if $C_t = F\Psi^\top\Psi + \frac{1}{\mu}FAA^\top$. Thus $F_t = C_t(\Psi^\top\Psi + \frac{1}{\mu}AA^\top)^\dagger$. The pseudo-inversion of the matrix can be pre-computed before iteration.

The larger μ is, the less effective is the constraint and thus the closer is the solution to the least-squares solution. In practice, μ is usually set to a small positive constant.

V. Empirical Results

In this section, we provide empirical results of learning an optimal control policy for cart-pole balancing using the AVI algorithm from the LS model and the constraint model.

A. Cart-pole Balancing

In this problem, the goal is to keep the pole with one end attached to the cart above the horizontal line ($|\vartheta| \le \pi/2$). The agent can choose between three actions to apply to the cart at each time step: pushing to the left (action 1) or right (action 3) in 50 Newton or applying no force (action 2). A Gaussian noise with zero mean and a standard deviation of 10 Newton is added to the force. If the pole is below the horizontal line the task fails and a reward -1 is given; a constant zero reward is given for the other states. The state variables are the angle and angular velocity of the pole, $[\theta, \dot{\theta}]$, which are both continuous. The discount factor is 0.9. No exploration was used.

Recall that for both the LS model and the constraint model the goal is to learn a matrix F such that $\hat{I} = \Psi F^\top$ approximates the identity matrix $I_{n\times n}$ well where n is the number of samples. We first tried the nine radial basis functions plus a constant feature by LSPI authors [4] for our AVI algorithms (with both the LS model and the L1 constraint model). For a state s, $\phi_i(s) = \exp(-||s - u_{i-1}||^2/2)$, $i = 1, 2, \ldots, 10$, where $u_0 = s$, and the other u_i are the points from the grid $\{-\pi/4, 0, \pi/4\} \times \{-1, 0, 1\}$ [4]. The algorithms did not perform well with these features. It turns out that the approximation of the identity matrix is poor for both models. For example, the LS approximation is shown in Figure 3 using about $1,300$ samples collected using a random policy by starting the pole from a random state near the state $[0, 0]$. The diagonal part is well approximated but the other part is noisy.

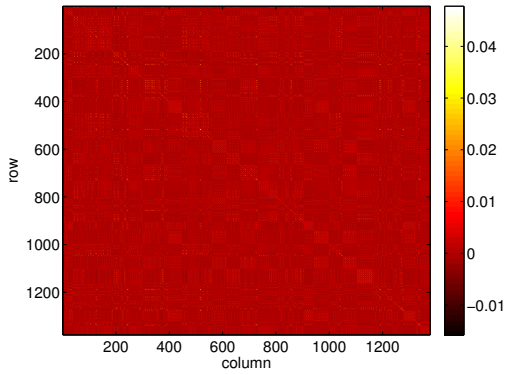


Fig. 3. The approximate identity matrix by the RBF features (LS fit).

To circumvent this problem, we used "tensor-product features". We first partitioned the state space using a grid and then pre-computed the RBF features inside each cell to provide generalization. As a result, both the LS model and the constraint model approximate the identity matrix well. For example, Figure 4 shows the LS approximation. In these two figures, we partitioned each state dimension into three parts. There are effectively three cells laid over the state space because six of them are all failure states (with $|\theta| > \pi/2$) whose feature vector is all zero. To illustrate the matrix better, we had sorted the samples according to the grid index that x_i belongs to and then according to the action a_i using a stable sorting algorithm. Because of the way the features are constructed, the approximate matrix contains only diagonal blocks and outside these blocks the values are strictly zero. The sorting operation ensures that the diagonal part of the approximate identity matrix is in the order of action blocks, each of which contains the smaller approximate identity matrices for the grids. The approximation is better with more partitions. Figure 5 shows the optimization solution using five partitions in each dimension. The ADMM algorithm was run with $\mu = 1.0$ and 30 iterations. The algorithm was fast and took 318 seconds for 30 iterations on a desktop with 1.7GHz Intel Core i7 and 8GB 1600 MHz DDR3.

In order to evaluate the performance of the normalized LS model and the constraint model, we conducted 30 independent runs of experiment. In each run, we collected a number of

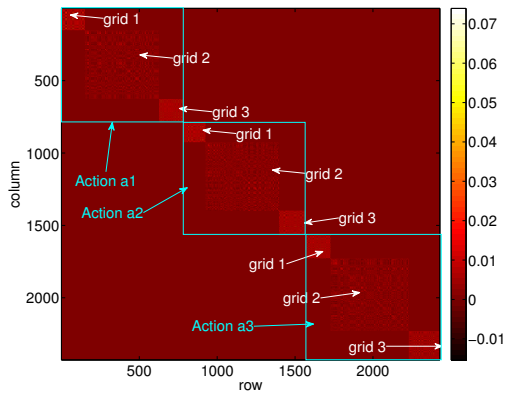


Fig. 4. The approximate identity matrix by the tensor-product features (LS fit using 3 partitions in each state dimension).

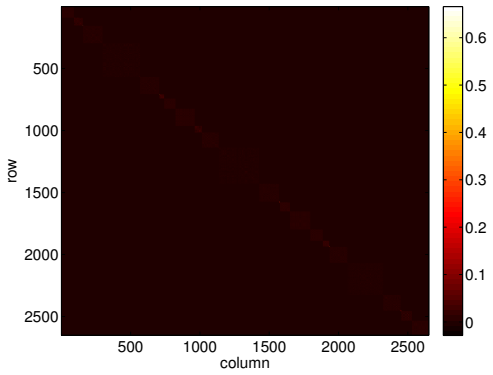


Fig. 5. The approximate identity matrix by the tensor-product features (The constrained optimization approximation using 5 partitions in each state dimension).

episodes of samples from the random policy. We learned the LS model and the constraint model, and then used them independently in AVI to compute an approximate optimal policy. The LS model was normalized by the L1 norm of each π. Each model was fed into the AVI procedure to produce a policy. We then evaluated each policy 100 times with up to 3000 steps in each evaluation. The averaged number of balanced steps was then used as a quality measure. Figure 6 shows the balanced steps by the policies of both methods. The constraint model is substantially better than the LS model.

VI. Conclusion

In this paper we proposed a framework called pseudo-MDPs which are more general than MDPs in that the transition kernel does not have to be a probability kernel. The advantage of pseudo-MDPs is that it allows one to solve a MDP from a broader class of approximate models. We provide a general AVI algorithm for pseudo-MDPs and theoretical guarantees as well as a generic error bound which recovers existing bounds. We propose two efficient approaches of learning a factored linear action model which constructs a pseudo-MDP.

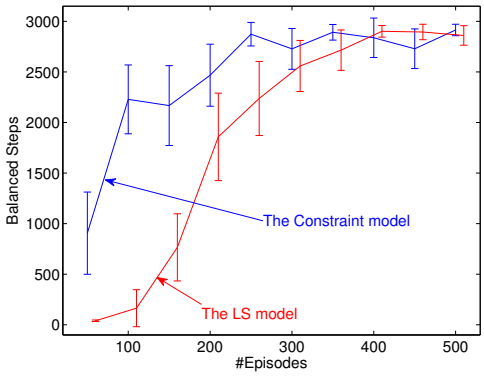


Fig. 6. The balanced steps of the pole for the cart-pole system by the AVI algorithms using the normalized LS model and the constraint model.

Results show that the approaches perform well; in addition, the constrained optimization approach is better than the least-squares approach.

References

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

[2] C. Szepesvári, *Algorithms for Reinforcement Learning*. Morgan and Claypool, July 2010.

[3] S. Grünewälder, G. Lever, L. Baldassarre, M. Pontil, and A. Gretton, "Modelling transition dynamics in MDPs with RKHS embeddings," *ICML*, pp. 535–542, 2012.

[4] M. Lagoudakis and R. Parr, "Least-squares policy iteration," *JMLR*, vol. 4, pp. 1107–1149, 2003.

[5] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic Programming*. Athena, 1996.

[6] B. Van Roy, "Performance loss bounds for approximate value iteration with state aggregation," *Math. Oper. Res.*, vol. 31, pp. 234–244, Feb. 2006.

[7] H. Yao and C. Szepesvári, "Approximate policy iteration with linear action models," *AAAI*, 2012.

[8] D. Ormoneit and S. Sen, "Kernel-based reinforcement learning," *Mach. Learn.*, vol. 49, pp. 161–178, Nov. 2002.

[9] Y. Ye, "The simplex method is strongly polynomial for the markov decision problem with a fixed discount rate," 2010.

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, pp. 1–122, Jan. 2011.

[11] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l1-ball for learning in high dimensions," *ICML*, pp. 272–279, 2008.