Learning Good State and Action Representations via Tensor Decomposition

Chengzhuo Ni
Princeton University
Princeton, NJ 08544, USA
chengzhuo.ni@princeton.edu

Anru R. Zhang
University of Wisconsin-Madison
Madison, WI 53706, USA
anruzhang@stat.wisc.edu

Yaqi Duan and Mengdi Wang
Princeton University
Princeton, NJ 08544, USA
yaqid@princeton.edu; mengdiw@princeton.edu

2021 IEEE International Symposium on Information Theory (ISIT) | 978-1-5386-8209-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISIT45174.2021.9518158

***Abstract*—The transition kernel of a continuous-state-action Markov decision process (MDP) admits a natural tensor structure. This paper proposes a tensor-inspired unsupervised learning method to identify meaningful low-dimensional state and action representations from empirical trajectories. The method exploits the MDP's tensor structure by kernelization, importance sampling and low-Tucker-rank approximation. This method can be further used to cluster states and actions respectively and find the best discrete MDP abstraction. We provide sharp statistical error bounds for tensor concentration and the preservation of diffusion distance after embedding.**

I. Introduction

State abstraction is a core problem at the heart of control and reinforcement learning (RL). In high-dimension RL, a naive grid discretization of the continuous state space often leads to exponentially many discrete states - an open challenge known as the curse of dimensionality. Having good state representations will significantly improve the efficiency of RL, by enabling the use of function approximation to better generalize knowledge from seen states to unseen states.

We say a state/action representation is "good", if it enables the use of function approximation to extrapolate and predict future value of unseen states. Suppose there is a representation allowing exact linear parametrization of the transition and value functions, then the sample complexity of RL reduces to depend linearly on d - the representation's dimension [1]–[4]. Even if exact parametrization is not possible, a good representation can be still useful for solving RL with approximation error guarantee (see discussions in [5], [6]). An important related problem is to strategically explore in online RL while learning state abstractions [7], [8]. Motivated by these advances, we desire methods that can learn good representations, for RL with high-dimensional state and action spaces, automatically from empirical data.

What further complicates the problem is the large action space. An action can be either a one-step decision or a sequence of multi-step decisions (known as *option*). States under different actions lead to very different dynamics. Although states and actions may admit separate low-dimensional structures, they are entangled with each other in sample trajectories. This necessitates the tensor approach to decouple actions from states, so that we can learn their abstractions respectively.

A. Our Approach

In this paper, we study the state and action abstraction of Markov decision processes (MDP) from a tensor decomposition view. We focus on the batch data setting. The Tucker decomposition structure of a transition kernel p provides natural abstractions of the state and action spaces. We illustrate the low-Tucker-rank property in a number of reduced-order MDP models, including the block MDP (aka hard aggregation), latent-state model (aka soft aggregation).

Suppose we are given state-action-state transition samples $\mathcal{D} = \{(s, a, s')\}$ from a long sample path generated by a behavior policy. Our objective is to identify a state embedding map and an action embedding map, which map the original state and action spaces (maybe continuous and high-dimensional) into low-dimensional representations, respectively. The embedding maps are desired to be maximally "predicative", by preserving a notion of kernelized diffusion distance that measures similarity between states in terms of their future dynamics.

To handle continuous state and action spaces, we use nonparametric function approximation with known kernel functions over the state and action spaces. By approximately decomposing the kernel into finitely many features, we are able to handle the continuous problem by estimating a transition tensor of finite dimensions. Next, we leverage importance sampling and low-rank tensor approximation to identify the desired state and action embedding maps. They yield "good" representations of states and actions that are useful for linear function approximation in RL. Further, these representations can be used to find the best discrete approximation to the MDP, and in particular, recover the latent structures of block MDP with high accuracy. To the best of knowledge, this paper makes the first attempt to learn low-rank representations for high-dimensional continuous Markov decision, with statistical guarantee. Figure 1 illustrates the main idea of our approach.

Contributions of this paper include:

- A tensor-inspired kernelized embedding method to learn low-dimensional state and action representations from empirical trajectories. The method exploits the MDP's tensor structure by importance sampling, mean embedding and low-rank approximation.
- Theoretical guarantee that the embedding maps largely preserve the "predictability" of states and actions in terms

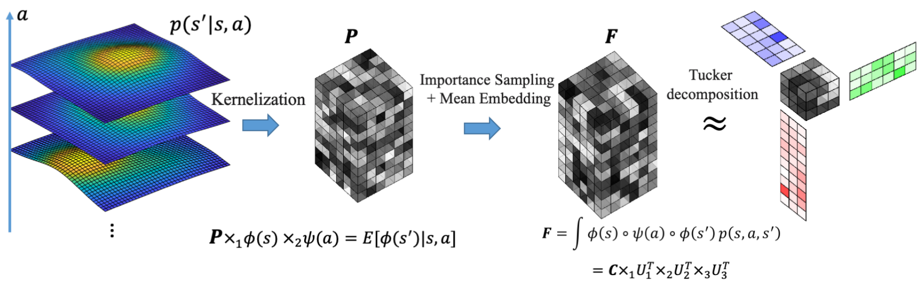

Fig. 1. An illustration of our tensor-inspired state and action embedding method.

of a kernelized diffusion distance, which is proved using a novel tensor concentration analysis.

- The numerical studies to corroborate our theoretical findings. The simulation results show the advantage of the proposed method over the baselines of vanilla and top r kernel PCA methods.

B. Related Literatures

Spectral and low-rank methods for dimension reduction have a long history. Our approach traces back to the diffusion map approach for manifold learning and graph analysis [9], which comes with a notion of diffusion distance that quantifies similarity between two nodes in a random walk. [10] extended the idea to systems driven by stochastic differential equations. [11] and [12], [13] studied how to infer dynamics of a system from leading spectrum of transition operator and find coresets of the state space.

The statistical theory of low-rank Markov model estimation received attention in recent years. [14] studied low-rank estimation of finite-state Markov chains. [15] studied the nonparametric estimation of transition kernel for continuous-state reversible Markov processes with exponentially decaying eigenvalues. [16] studied kernelized state embedding and statistical estimation of metastable clusters. These results only apply to Markov processes.

In control theory and RL, state aggregation is a long known approach for reducing the complexity of the state space; see e.g., [17]–[21]. Representation learning methods were proposed that uses diagonalization or dilation of some Laplacian operator as a surrogate of the transition operator; see e.g. [22]–[25]. See [26] for a review. These methods typically require prior knowledge about structures of the problem such as the transition function.

General methods for tensor decomposition and low-rank approximation have been studied in the applied math, statistics, and computer science literature, including the high-order singular value decomposition (HOSVD) [27], high-order orthogonal iteration (HOOI) [28], best low-rank approximation [29], [30], sketched-based algorithms [31], power iteration, k-means power iteration [32], etc. The readers are also referred to surveys [33], [34].

C. Markov Decision Process

An instance of a Markov decision process can be specified by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r)$, where $\mathcal{S}$ and $\mathcal{A}$ are state and action spaces, p is the transition probability kernel, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the one-step reward function. At each step t, suppose the current state is s_t. If the agents choose an action a_t, they will receive an instant reward $r(s_t, a_t) \in [0, 1]$ and system's state will transit to s_{t+1} according to the probability distribution $p(\cdot|s_t, a_t)$. A policy π is a rule for choosing actions based on states, where $\pi(\cdot|s)$ is a probability distribution over $\mathcal{A}$ conditioned on $s \in \mathcal{S}$. Under a given policy, the transition of the MDP will reduce to a Markov chain, whose transition kernel is denoted by p^π where $p^\pi(s'|s) = p^{1,\pi}(s'|s) = \int_{\mathcal{A}} \pi(a|s)p(s'|s, a)da$. Based on that, we define the t-step transition kernel $p^{t,\pi}(\cdot|s)$ inductively by $p^{t,\pi}(\cdot|s) = \int p^{t-1,\pi}(s'|s)p^\pi(\cdot|s')ds'$. And we further use ν^π to denote the invariant distribution of that Markov chain. Define the worst-case mixing time [35, Page 55] as

$$t_{mix} = \max_\pi \min \big\{t \big| \|p^{t',\pi}(\cdot|s_0) - \nu^\pi\|_{TV} \le 1/4, \\ \forall s_0 \in \mathcal{S}, t' \ge t\big\},$$

where $\|\cdot\|_{TV}$ denotes the total variation distance.

D. Tensor and Tucker Decomposition

For a general tensor $\boldsymbol{X} \in \mathbb{R}^{p_1\times p_2\times\cdots p_N}$, we denote $\boldsymbol{X} \times_n \boldsymbol{U}$ as the product between $\boldsymbol{X}$ and a matrix $\boldsymbol{U} \in \mathbb{R}^{p_n\times q}$ on the n^{th} dimension, which is of size $p_1 \times \ldots \times p_{n-1} \times q \times p_{n+1} \times \ldots \times p_N$. Each element of $\boldsymbol{X} \times_n \boldsymbol{U}$ is defined as $(\boldsymbol{X} \times_n \boldsymbol{U})_{i_1\ldots i_{n-1}ji_{n+1}\ldots i_N} = \sum_{i=1}^{p_n} \boldsymbol{X}_{i_1\ldots i_{n-1}ii_{n+1}\ldots i_N}\boldsymbol{U}_{ij}$. We denote by $\mathcal{M}_k(\boldsymbol{X}) \in \mathbb{R}^{p_k\times\prod_{i\neq k}p_i}$ the factor-k matricization (or flattening) of $\boldsymbol{X}$. The Tucker decomposition of $\boldsymbol{X}$ is of the form $\boldsymbol{X} = \boldsymbol{G} \times_1 \boldsymbol{U}_1 \times_2 \ldots \times_N \boldsymbol{U}_N$, where $\boldsymbol{G} \in \mathbb{R}^{q_1\times\ldots\times q_N}$ is a smaller core tensor. In particular, we call the smallest size of $\boldsymbol{G}$ the Tucker-rank of $\boldsymbol{X}$. Rigorously, we define Tucker-Rank$(\boldsymbol{X}) = (R_1, R_2, \ldots, R_N)$, where $R_k = \text{Rank}(\mathcal{M}_k(\boldsymbol{X}))$. The inner product between two tensors $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{p_1\times p_2\times\cdots p_N}$ is defined as $\langle\boldsymbol{X}, \boldsymbol{Y}\rangle = \sum_{i_1=1}^{p_1}\sum_{i_2=1}^{p_2}\cdots\sum_{i_N=1}^{p_N} \boldsymbol{X}_{i_1i_2\cdots i_N}\boldsymbol{Y}_{i_1i_2\cdots i_N}$. The spectral norm and Frobenius norm of a tensor $\boldsymbol{X} \in \mathbb{R}^{p_1\times p_2\times\ldots\times p_N}$ are defined as $\|\boldsymbol{X}\|_\sigma = \sup_{\|u_i\|=1, 1\le i\le N}\langle\boldsymbol{X}, u_1 \circ u_2 \circ \ldots \circ u_N\rangle$, $\|\boldsymbol{X}\|_F = \sqrt{\langle\boldsymbol{X}, \boldsymbol{X}\rangle}$.

II. A Tensor View of Markov Decision Process

Consider a continuous-state MDP with the transition kernel p, where each $p(\cdot|s, a)$ is a conditional transition density function. We adopt a tensor view to exploit structures of p for abstractions of state and action spaces. We handle the continuous state and action spaces using kernel function approximation. Suppose we have a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_S$ for functions over states and an RKHS $\mathcal{H}_A$ for functions over actions. We make the assumption that the MDP's transition kernel p can be represented in these function spaces.

Assumption 1. *Let $\mathbb{P}$ be the transition operator of p, i.e., $(\mathbb{P}f)(s,a) = \int p(s'|s,a)f(s')ds'$. Assume that Tucker-Rank$(\mathbb{P}) \le (r,l,m)$ [1], and $\mathbb{P}f \in \mathcal{H}_S \times \mathcal{H}_A, \forall f \in \mathcal{H}_S$.*

Here, the low-Tucker rankness assumption captures the structure that state/action space can be compressed into a lower-dimensional space while preserving the dynamics. This assumption naturally holds in many well-known reinforcement learning models, such as soft state aggregation [19], [36], [37], block MDP [38], rich-observation MDP [7], contextual MDP [39], linear/factor MDP [4], kernel MDP [40], [41]. We remark that the tensor rank is determined solely by the transition model p (i.e., the environment), regardless of the reward r.

We give two basic examples of low rank MDPs below.

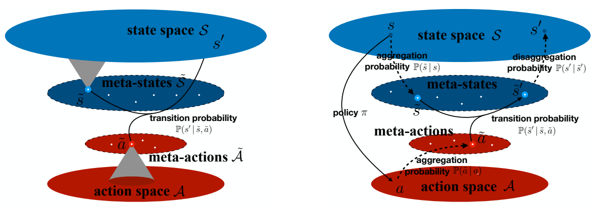

Fig. 2. Left: Block MDP (aka hard aggregation); Right: Latent-state-action MDP (aka soft aggregation).

Example 1 (Block MDP (Hard Aggregation, , Fig. 2 Left))**.** Let $\tilde{\mathcal{S}}$ and $\tilde{\mathcal{A}}$ be finite sets. Suppose there exists state and action abstractions $f : \mathcal{S} \mapsto \tilde{\mathcal{S}}$ and $g : \mathcal{S} \mapsto \tilde{\mathcal{A}}$ such that

$$p(\cdot|s,a) = p(\cdot|s',a') \text{ if } f(s) = f(s'), g(a) = g(a')$$

Then p has Tucker rank at most $(|\tilde{\mathcal{S}}|, |\tilde{\mathcal{A}}|, |\mathcal{S}|)$.

Example 2 (Latent-State-Action MDP (Soft Aggregation, Fig. 2 Right))**.** Given an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r)$, we say $\mathcal{M}$ has an (r,l,m)-latent variable model if there exist a latent state-action-state stochastic process $\{\tilde{s}_t, \tilde{a}_t, \tilde{s}'_t\} \subseteq \tilde{\mathcal{S}} \times \tilde{\mathcal{A}} \times \tilde{\mathcal{S}}'$, with $|\tilde{\mathcal{S}}| = r, |\tilde{\mathcal{A}}| = l, |\tilde{\mathcal{S}}'| = m$, such that

$$\mathbb{P}(\tilde{s}_t, \tilde{a}_t | s_1, a_1, \ldots, s_t, a_t) = \mathbb{P}(\tilde{s}_t | s_t)\mathbb{P}(\tilde{a}_t | a_t),$$
$$\mathbb{P}(\tilde{s}'_t | s_1, a_1, \ldots, s_t, a_t, \tilde{s}_t, \tilde{a}_t) = \mathbb{P}(\tilde{s}'_t | \tilde{s}_t, \tilde{a}_t),$$
$$\mathbb{P}(s_{t+1} | s_1, a_1, \ldots, s_t, a_t, \tilde{s}_t, \tilde{a}_t, \tilde{s}'_t) = \mathbb{P}(s_{t+1} | \tilde{s}'_t).$$

In this case, one can verify that p has Tucker rank (r,l,m).

In the remainder of the paper, we assume without loss of generality that the state and action kernel spaces admit finitely many known basis functions, which we refer to as state features $\phi(s) \in \mathbb{R}^{d_S}$ and action features $\psi(a) \in \mathbb{R}^{d_A}$. This is a rather mild assumption: Even if we do not know the basis function but are only given kernel functions K_1 and K_2 for $\mathcal{H}_S$ and $\mathcal{H}_A$. According to [42], we can always generate finitely many random features to approximately span these kernel spaces such that $K_1(s,s') \approx \sum_{i=1}^{d_S} \phi_i(s)^\top \phi_i(s')$ and $K_2(a,a') \approx \sum_{i=1}^{d_A} \psi_i(a)^\top \psi_i(a')$. Also note that our approach applies to *arbitrary* state and action spaces, as long as they come with appropriate kernel functions. Although p is infinitely dimensional, we use the given kernel spaces and represent p with a finite-dimensional tensor. In particular, Assumption 1 implies the following tensor linear model:

Lemma 1 (Conditional transition tensor and linear model)**.** *Suppose Assumption 1 holds. There exists a tensor $\mathbf{P} \in \mathbb{R}^{d_S \times d_A \times d_S}$ such that Tucker-Rank$(\boldsymbol{P}) \le (r,l,m)$ and*

$$\mathbf{P} \times_1 \phi(s) \times_2 \psi(a) = \mathbb{E}[\phi(s')|s,a], \qquad \forall s, a.$$

Before closing this section, we give an illustrative example to show the advantage of utilizing tensor MDP formulation as opposed to the matrix ones.

Example 3. Consider $\mathcal{A} = \{1,2\}$, $\mathcal{S} = \{1,2,3,4\}$. Construct the MDP transition tensor $\boldsymbol{P}$ as

$$\boldsymbol{P}_{\cdot 1 \cdot} = \begin{bmatrix} 1/6 & 1/6 & 1/3 & 1/3 \\ 1/6 & 1/6 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/6 & 1/6 \\ 1/3 & 1/3 & 1/6 & 1/6 \end{bmatrix}, \boldsymbol{P}_{\cdot 2 \cdot} = \begin{bmatrix} 1/3 & 1/3 & 1/6 & 1/6 \\ 1/3 & 1/3 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/3 & 1/3 \\ 1/6 & 1/6 & 1/3 & 1/3 \end{bmatrix}.$$

Then, $\boldsymbol{P} = \boldsymbol{C} \times_1 \boldsymbol{U}^\top \times_3 \boldsymbol{U}^\top$ for

$$\boldsymbol{U} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, \boldsymbol{C}_{\cdot 1 \cdot} = \begin{bmatrix} 1/6 & 1/3 \\ 1/3 & 1/6 \end{bmatrix}, \boldsymbol{C}_{\cdot 2 \cdot} = \begin{bmatrix} 1/3 & 1/6 \\ 1/6 & 1/3 \end{bmatrix}$$

and the state-space is aggregatable into two latent states: $\{1,2\}$ and $\{3,4\}$. Consider a random policy: $\pi(a|s) = 1/2$ for $a = 1,2$. Without taking into account the tensor structure induced by the policy, one can check that the state transitions $\{s_0, s_1, \ldots\}$ form a Markov process with the following transition matrix

$$\tilde{\boldsymbol{P}} = (1/2)\boldsymbol{P}_{\cdot 1 \cdot} + (1/2)\boldsymbol{P}_{\cdot 2 \cdot} = (1/4) \cdot 1_{4 \times 4}.$$

Since the latent state partition is "averaged out" by any matrix methods, it is not feasible to extract the latent state information merely from the state transitions $\{s_0, s_1, \ldots\}$. In contrast, the proposed tensor formulation, which preserves the original state-action-state information, allows a reliable latent state recovery.

III. Tensor-Inspired State and Action Embedding Learning

In this section, we develop a tensor-inspired representation learning method, that embeds states and actions into decoupled low-dimensional spaces. Next, we will develop the method step by step, and provide theoretical guarantees.

A. Tensor MDP Mean Embedding by Importance Sampling

Suppose we have a batch dataset of state-action samples.

Assumption 2. *The data $\mathcal{D} = \{(s,a,s')\}$ consists of state-action-state transitions from a single sample path generated by a known behavior policy $\bar{\pi}$.*

[1] We define the Tucker-rank of an operator as follows: there exists $c_{ijk} \in \mathbb{R}$ and functions $u_i, w_k \in \mathcal{H}_S, v_j \in \mathcal{H}_A, i \in [r], j \in [l], k \in [m]$, such that $(\mathbb{P}f)(s,a) = \sum_{i=1}^{r} \sum_{j=1}^{l} \sum_{k=1}^{m} c_{ijk} u_i(s) v_j(a) \langle f, w_k \rangle_{\mathcal{H}_S}$

Let ξ be the stationary state distribution of the sample path under policy $\bar{\pi}$. Let η be a positive probability measure over the action space. Consider the tensor mean embedding

$$\boldsymbol{F} = \int \phi(s) \circ \psi(a) \circ \phi(s') p(s,a,s') ds da ds' \in \mathbb{R}^{d_S \times d_A \times d_S},$$

where $p(s,a,s') = p(s'|s,a)\xi(s)\eta(a)$.

Lemma 2. *Assumption 1 implies Tucker-Rank($\boldsymbol{F}$) $\leq (r,l,m)$.*

We estimate the mean embedding tensor $\boldsymbol{F}$ by importance sampling:

$$\bar{\boldsymbol{F}} = n^{-1} \textstyle\sum_{i=1}^n \eta(a_i)/\bar{\pi}(a_i|s_i) \cdot \phi(s_i) \circ \psi(a_i) \circ \phi(s_i'). \quad (1)$$

The mean embedding tensor $\boldsymbol{F}$ is related to the transition tensor $\boldsymbol{P}$ through a simple relation.

Lemma 3 (Relation between $\boldsymbol{P}$ and $\boldsymbol{F}$)**.** *When $\{\psi_i(\cdot)\}_{i=1}^{d_A}$ forms a set of orthogonal basis with respect to $L^2(\eta)$, we have $\boldsymbol{P} = \boldsymbol{F} \times_1 \boldsymbol{\Sigma}^{-1}$, where $\boldsymbol{\Sigma} = \int \xi(s)\phi(s)\phi(s)^\top ds$.*

B. Low-Rank Estimation of Transition Tensor

We estimate a low-rank approximation to $\boldsymbol{F}$ by solving:

$$\begin{aligned} &\hat{\boldsymbol{F}} = \text{argmin } \|\boldsymbol{Q} - \bar{\boldsymbol{F}}\|_\sigma, \\ &\text{subject to Tucker-Rank}(\boldsymbol{Q}) \leq (r,l,m). \end{aligned} \quad (2)$$

and estimate the transition operator $\boldsymbol{P}$ by $\hat{\boldsymbol{P}} = \hat{\boldsymbol{F}} \times_1 \hat{\boldsymbol{\Sigma}}^{-1}$, where $\hat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^n \phi(s_i)\phi^\top(s_i)$. Define

$$\begin{aligned} &K_{max} = \max\{\sup_s K_1(s,s), \sup_a K_2(a,a)\}, \\ &\bar{\mu} = \|\mathbb{E}[K_1(S,S)\phi(S)\phi(S)^\top]\|_\sigma, \kappa = \sup_{s\in\mathcal{S}, a\in\mathcal{A}} \eta(a)/\pi(a|s), \\ &\bar{\lambda} = \sup_{\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}} \mathbb{E}_{\xi\circ\eta\circ p(\cdot|\cdot)}[[(\boldsymbol{u}^\top\phi(S))(\boldsymbol{v}^\top\psi(A))(\boldsymbol{w}^\top\phi(S'))]^2], \\ &\text{where } \boldsymbol{u}, \boldsymbol{w} \in S^{d_S - 1}, \boldsymbol{v} \in S^{d_A-1}. \end{aligned}$$

Theorem 1 (Low-rank estimation of the transition tensor $\boldsymbol{P}$)**.** *Suppose Assumptions 1-2 hold. Suppose ψ is orthonormal with respect to $L^2(\eta)$, and*

$$\begin{aligned} \frac{n/t_{mix}}{(\log(n/t_{mix}))^2} \geq 1024\left(\|\boldsymbol{\Sigma}^{-1}\|_\sigma^2\bar{\mu} + \frac{K_{max}^2}{\bar{\mu}} + \frac{\kappa K_{max}^3}{\bar{\lambda}}\right) \\ \cdot\left(\log\frac{2t_{mix}}{\delta} + 8(d_S + d_A)\right), \end{aligned}$$

then with probability $1-\delta$, we have

$$\begin{aligned} &\|\boldsymbol{P} - \hat{\boldsymbol{P}}\|_\sigma \\ \leq &256\|\boldsymbol{\Sigma}^{-1}\|_\sigma \sqrt{\frac{\bar{\lambda}(\log\frac{2t_{mix}}{\delta} + d_S + d_A)(\kappa + \bar{\mu}\|\boldsymbol{\Sigma}^{-1}\|_\sigma^2)}{(n/t_{mix})\log^{-2}(n/t_{mix})}}. \end{aligned}$$

The derivation of $\hat{\boldsymbol{P}}$ also provides a tractable way to estimate $\mathbb{E}[\phi(s')|s,a]$ by $\hat{\mathbb{E}}[\phi(s')|s,a] := \hat{\boldsymbol{P}} \times_1 \phi(s) \times_2 \psi(a)$. We have the following guarantee on the estimation error: $\|\hat{\mathbb{E}}[\phi(s')|s,a] - \mathbb{E}[\phi(s')|s,a]\| \leq K_{max}\|\hat{\boldsymbol{P}} - \boldsymbol{P}\|_\sigma$.

a) Rank selection: Our theory assumes prior knowledges of the tensor rank. In practice, it is common to tune the rank parameters by checking the elbow in the scree plot and using cross validation (see discussions in the classical literature on PCA, e.g., [43]). In theory, rank estimation is hard unless one makes additional strong assumptions, like that the eigengap is bounded from below.

b) Computation: Finding the exact optimum of (2) can be computationally intense in general [44]. In practice, we can apply classic tensor decomposition algorithms, such as higher-order orthogonal iteration (HOOI) [28], high-order SVD [27], sequential-HOSVD [45], gradient descent [46], to find an approximate solution to (2). In particular, the statistical optimality of tensor power iterations, e.g. HOOI and HOSVD (Appendix A), have been justified in some special cases [30]. We expect these approximations also work for our problems, which is later validated in our experiment.

C. Learning State and Action Embeddings

Next, we show how to embed states and actions to low-dimensional representations to be maximally "predictive." Consider a kernelized diffusion distance of the MDP, which measures similarity in terms of future dynamics restricted to a function class:

$$\text{dist}[(s_1,a_1),(s_2,a_2)] = \sup_{\|f\|_{\mathcal{H}_S}\leq 1} |\mathbb{E}[f(s')|s_1,a_1] - \mathbb{E}[f(s')|s_2,a_2]|.$$

This distance quantifies how well one can generalize the predicted value at a seen state-action pair (s,a) to a new (s',a'). Under the low-tensor-rank assumption, we have $\boldsymbol{P} = \boldsymbol{C} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top$, where $\boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3$ are columnwisely orthonormal matrices. Then we can define the *kernelized state diffusion map*, *kernelized action diffusion map* and their joint map as

$$f(\cdot) := \boldsymbol{U}_1^\top\phi(\cdot), \; g(\cdot) := \boldsymbol{U}_2^\top\psi(\cdot), \; \Phi(s,a) := \boldsymbol{C} \times_1 f(s) \times_2 g(a).$$

It follows that $\text{dist}[(s,a),(s',a')] = \|\Phi(s,a) - \Phi(s',a')\|$, if ϕ is a collection of orthonormal basis functions of $\mathcal{H}_S$. Motivated by the preceding analysis, we propose to estimate state and action embedding maps based on the tensor estimator. After we obtain $\hat{\boldsymbol{P}}$, we can simply find the corresponding state and action embedding maps from factors of its Tucker decomposition

$$\hat{\boldsymbol{P}} = \hat{\boldsymbol{C}} \times_1 \hat{\boldsymbol{U}}_1^\top \times_2 \hat{\boldsymbol{U}}_2^\top \times_3 \hat{\boldsymbol{U}}_3^\top.$$

Now we have obtained the state embedding map $\hat{f}$ and the action embedding map $\hat{g}$. Accordingly, we define the joint state-action embedding and the empirical embedding distance as

$$\begin{aligned} &\hat{\Phi}(s,a) = \hat{\boldsymbol{C}} \times_1 \hat{f}(s) \times_2 \hat{g}(a), \; \widehat{\text{dist}}[(s,a),(s',a')] \\ =&\|\hat{\Phi}(s,a) - \hat{\Phi}(s',a')\|. \end{aligned}$$

Algorithm 1 Learning State and Action Embedding Maps

1: **Input**: $\{(s_i,a_i,s_i')\}_{i=1}^n, (r,l,m)$
2: Calculate $\bar{\boldsymbol{F}} = \frac{1}{n}\sum_{i=1}^n \frac{\eta(a_i)}{\pi(a_i|s_i)}\phi(s_i)\circ\psi(a_i)\circ\phi(s_i')$, get $\hat{\boldsymbol{F}}$ as the low-rank approximation of $\bar{\boldsymbol{F}}$ using (2)
3: Calculate $\hat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^n \phi(s_i)\phi^\top(s_i)$, $\hat{\boldsymbol{P}} = \hat{\boldsymbol{F}} \times_1 \hat{\boldsymbol{\Sigma}}^{-1}$
4: Let $\boldsymbol{P}_1 = \hat{\boldsymbol{P}}$. For $k = 1,2,3$, derive $\hat{\boldsymbol{U}}_k$ from the SVD $\mathcal{M}_k(\boldsymbol{P}_k) = \hat{\boldsymbol{U}}_k\boldsymbol{\Lambda}_k\boldsymbol{V}_k^\top$, and let $\boldsymbol{P}_{k+1} = \boldsymbol{P}_k \times_k \hat{\boldsymbol{U}}_k$.
5: **Output**:
State and action embedding maps $\hat{f} : s \mapsto \hat{\boldsymbol{U}}_1^\top\phi(s), \hat{g} : a \mapsto \hat{\boldsymbol{U}}_2^\top\psi(a)$; Core transition tensor $\hat{\boldsymbol{C}} = \boldsymbol{P}_4$.

Theorem 2 (Embedding error bound)**.** *Let Assumptions 1-2 hold. Suppose ϕ is an orthogonal basis of $\mathcal{H}_S$, and ψ is*

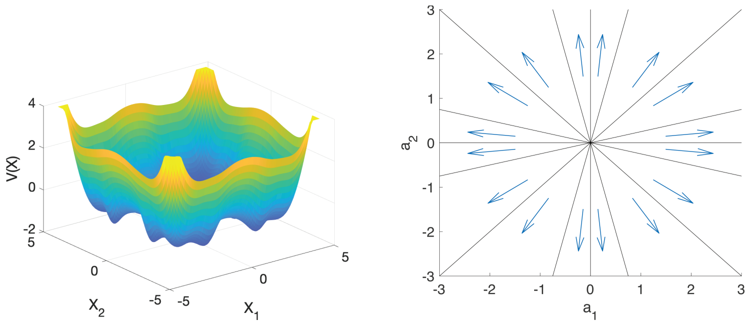

Fig. 3. Left: Potential function $V(\cdot)$; Right: Block-wise control function $F(\cdot)$. The action space has 16 blocks, and in each block $F(\cdot)$ is a constant drift vector (see the arrows).

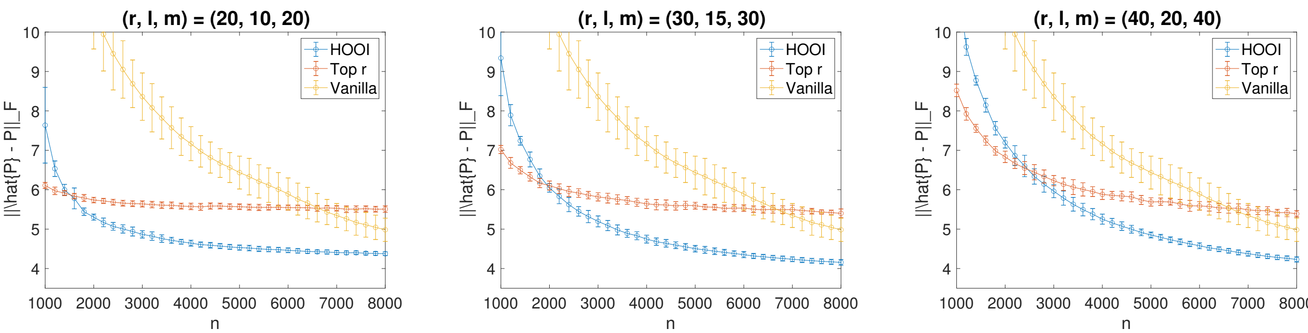

Fig. 4. Low-tensor-rank estimation of $\boldsymbol{P}$, compared with baseline methods.

orthogonal w.r.t $L^2(\eta)$, then we can find an orthogonal matrix $\boldsymbol{O}$, such that

$$\|\hat{\Phi}(s,a) - \boldsymbol{O}\Phi(s,a)\| \leq \epsilon,$$
$$|\widehat{dist}[(s,a),(s',a')] - dist[(s,a),(s',a')]| \leq 2\epsilon, \forall s,a,s',a'$$

where ϵ is controlled by the low-rank estimation error, $\epsilon := K_{max}\left(1+\sqrt{2}\|\boldsymbol{P}\|_\sigma/\sigma\right)\|\hat{\boldsymbol{P}} - \boldsymbol{P}\|_\sigma$, and $\sigma := \sup_{\|w\|\leq 1}\sigma_m(\boldsymbol{P}\times_1 w)$, where σ_m denotes the m-th singular value of a matrix.

a) Advantage of tensor method: As an alternative, one could ignore the tensor structure and treat the state and action jointly, yielding a low-dimensional representation for the pair (s,a) directly. This approach may be favorable if the (s,a) has a very simple joint structure. However, the tensor approach may be significantly more sample efficient if s and a admit *separate* low-dimensional structures. To see this, suppose the state and action features have dimensions d_S and d_A before embedding. Also assume the Tucker rank (r,l,m) are constants for simplicity. By treating (s,a) jointly and ignoring the tensor structure, one would need $\tilde{\Omega}(d_A d_S)$ (i.e., the matrix dimension) samples to reliably recover the low-dimensional structure. In comparison, our tensor-based approach requires only $\tilde{\Omega}(d_X + d_A)$ samples as indicated by Theorem 1.

IV. Numerical Experiment

We test our approach on a particular MDP derived from a controlled stochastic process. Let the state and action spaces be both $\mathbb{R}^2$. Suppose the state-action pair at step k is (s_k, a_k). Then the next state s_{k+1} is set to be $X_{\tau(k+1)}$ for some $\tau > 0$, where X_t is the solution of the SDE:

$$dX_t = -[\nabla V(X_t) + F(a_k)]dt + \sqrt{2}dB_t, k\tau \leq t \leq (k+1)\tau,$$

where $V(\cdot)$ is a wavy potential function, $F(\cdot)$ is a block-wise constant function (Figure 3), B_t is the standard Brownian motion. Let the behavior policy be always choosing a from a standard normal distribution. In the experiment, we use the Gaussian kernels $K_1(x,y) = K_2(x,y) = \frac{1}{2\pi\sigma^2}\exp\{-\frac{\|x-y\|^2}{2\sigma^2}\}$. The features are obtained by generating N_s (or N_a) random Fourier features $h = [h_1, h_2, \cdots, h_{N_s}]$ such that $K(x,y) \approx \sum_{i=1}^{N_s(N_a)} h_i(x)h_i(y)$. The action features are then orthogonalized with respect to $L^2(\eta)$. We also choose $\tau = 0.1, \sigma = 0.5, N_s = 100, N_a = 50$.

We investigate the efficiency of estimating $\boldsymbol{P}$ via the proposed method and compare with two baseline methods: (1) The vanilla method, which directly estimates the transition tensor by $\hat{\boldsymbol{P}} = \bar{\boldsymbol{F}}\times_1 \bar{\boldsymbol{\Sigma}}^{-1}$ without any low-rank approximation; (2) The "top r" method, whose the procedure is: i) calculate the top r (or l, m) principle components of the sample covariance per mode; ii) project features onto the subspace spanned by the top principle components; iii) estimate the transition tensor via the vanilla method (discussed above) in the space of projected features. Fig. 4 visualizes the estimation errors of these methods with different choices of (r,l,m), where errors are averaged over five independent runs. We observe that, for most of the time, our method consistently outperforms the baselines. Note that the top r method performs slightly better when n is very small, because in this case data is too small to get meaningful estimate of $\boldsymbol{P}$.

References

[1] M. G. Lagoudakis and R. Parr, "Least-squares policy iteration," *Journal of machine learning research*, vol. 4, no. Dec, pp. 1107–1149, 2003.

[2] A. Zanette, A. Lazaric, M. J. Kochenderfer, and E. Brunskill, "Limiting extrapolation in linear approximate value iteration," in *Advances in Neural Information Processing Systems*, 2019, pp. 5616–5625.

[3] L. Yang and M. Wang, "Sample-optimal parametric q-learning using linearly additive features," in *International Conference on Machine Learning*, 2019, pp. 6995–7004.

[4] C. Jin, Z. Yang, Z. Wang, and M. I. Jordan, "Provably efficient reinforcement learning with linear function approximation," *arXiv preprint arXiv:1907.05388*, 2019.

[5] S. S. Du, S. M. Kakade, R. Wang, and L. F. Yang, "Is a good representation sufficient for sample efficient reinforcement learning?" *arXiv preprint arXiv:1910.03016*, 2019.

[6] T. Lattimore and C. Szepesvari, "Learning with good feature representations in bandits and in rl with a generative model," *arXiv preprint arXiv:1911.07676*, 2019.

[7] S. S. Du, A. Krishnamurthy, N. Jiang, A. Agarwal, M. Dudík, and J. Langford, "Provably efficient rl with rich observations via latent state decoding," *arXiv preprint arXiv:1901.09018*, 2019.

[8] D. Misra, M. Henaff, A. Krishnamurthy, and J. Langford, "Kinematic state abstraction and provably efficient rich-observation reinforcement learning," *arXiv preprint arXiv:1911.05815*, 2019.

[9] S. Lafon and A. Lee, "Diffusion maps and coarse-graining: A uni ed framework for dimensionality reduction, graph partitioning, and data set parameterization," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1393–1403, 2006.

[10] R. R. Coifman, I. G. Kevrekidis, S. Lafon, M. Maggioni, and B. Nadler, "Diffusion maps, reduction coordinates, and low dimensional representation of stochastic systems," *SIAM Journal on Multiscale Modeling and Simulation*, vol. 7, no. 2, pp. 852–864, 2008.

[11] C. Schütte, F. Noe, J. Lu, M. Sarich, and E. Vanden-Eijnden, "Markov state models based on milestoning," *The Journal of Chemical Physics*, vol. 134, no. 20, p. 204105, 2011.

[12] S. Klus, P. Koltai, and C. Schütte, "On the numerical approximation of the perron–frobenius and koopman operator," *Journal of Computational Dynamics*, vol. 3, no. 1, pp. 51–79, 2016.

[13] S. Klus, I. Schuster, and K. Muandet, "Eigendecompositions of transfer operators in reproducing kernel hilbert spaces," *arXiv preprint arXiv:1712.01572*, 2018.

[14] A. Zhang and M. Wang, "Spectral state compression of markov processes," *IEEE Transactions on Information Theory*, vol. to appear, 2019.

[15] M. Löffler and A. Picard, "Spectral thresholding for the estimation of markov chain transition operators," *arXiv preprint arXiv:1808.08153*, 2018.

[16] Y. Sun, Y. Duan, H. Gong, and M. Wang, "Learning low-dimensional state embeddings and metastable clusters from time series data," in *Advances in Neural Information Processing Systems*, 2019, pp. 4563–4572.

[17] A. W. Moore, "Variable resolution dynamic programming: Efficiently learning action maps in multivariate real-valued state-spaces," in *Machine Learning Proceedings 1991*. Elsevier, 1991, pp. 333–337.

[18] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic programming*. Athena Scientific, Belmont, MA, 1996.

[19] S. P. Singh, T. Jaakkola, and M. I. Jordan, "Reinforcement learning with soft state aggregation," in *Advances in neural information processing systems*, 1995, pp. 361–368.

[20] J. N. Tsitsiklis and B. Van Roy, "Feature-based methods for large scale dynamic programming," *Machine Learning*, vol. 22, no. 1-3, pp. 59–94, 1996.

[21] Z. Ren and B. H. Krogh, "State aggregation in markov decision processes," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 4. IEEE, 2002, pp. 3819–3824.

[22] J. Johns and S. Mahadevan, "Constructing basis functions from directed graphs for value function approximation," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 385–392.

[23] S. Mahadevan, "Proto-value functions: Developmental reinforcement learning," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 553–560.

[24] R. Parr, C. Painter-Wakefield, L. Li, and M. Littman, "Analyzing feature generation for value-function approximation," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 737–744.

[25] M. Petrik, "An analysis of laplacian methods for value function approximation in mdps." in *IJCAI*, 2007, pp. 2574–2579.

[26] S. Mahadevan *et al.*, "Learning representation and control in markov decision processes: New frontiers," *Foundations and Trends® in Machine Learning*, vol. 1, no. 4, pp. 403–565, 2009.

[27] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[28] ——, "On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

[29] E. Richard and A. Montanari, "A statistical model for tensor pca," in *Advances in Neural Information Processing Systems*, 2014, pp. 2897–2905.

[30] A. Zhang and D. Xia, "Tensor svd: Statistical and computational limits," *IEEE Transactions on Information Theory*, vol. 64, no. 11, pp. 7311–7338, 2018.

[31] Z. Song, D. Woodruff, and H. Zhang, "Sublinear time orthogonal tensor decomposition," in *Advances in Neural Information Processing Systems*, 2016, pp. 793–801.

[32] A. Anandkumar, R. Ge, and M. Janzamin, "Guaranteed non-orthogonal tensor decomposition via alternating rank-1 updates," *arXiv preprint arXiv:1402.5180*, 2014.

[33] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[34] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015.

[35] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov chains and mixing times*. American Mathematical Soc., 2009.

[36] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 2007.

[37] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[38] K. Azizzadenesheli, A. Lazaric, and A. Anandkumar, "Reinforcement learning in rich-observation mdps using spectral methods," *arXiv preprint arXiv:1611.03907*, 2016.

[39] N. Jiang, A. Krishnamurthy, A. Agarwal, J. Langford, and R. E. Schapire, "Contextual decision processes with low bellman rank are pac-learnable," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1704–1713.

[40] D. Ormoneit and P. Glynn, "Kernel-based reinforcement learning in average-cost problems," *IEEE Transactions on Automatic Control*, vol. 47, no. 10, pp. 1624–1636, 2002.

[41] S. R. Chowdhury and A. Gopalan, "Online learning in kernelized markov decision processes," *arXiv preprint arXiv:1805.08052*, 2018.

[42] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in neural information processing systems*, 2008, pp. 1177–1184.

[43] I. T. Jolliffe, "Principal components in regression analysis," in *Principal component analysis*. Springer, 1986, pp. 129–155.

[44] V. De Silva and L.-H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1084–1127, 2008.

[45] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen, "A new truncation strategy for the higher-order singular value decomposition," *SIAM Journal on Scientific Computing*, vol. 34, no. 2, pp. A1027–A1052, 2012.

[46] R. Han, R. Willett, and A. Zhang, "An optimal statistical and computational framework for generalized tensor estimation," *arXiv preprint arXiv:2002.11255*, 2020.

[47] R. Vershynin, *High-Dimensional Probability*. Cambridge University Press (to appear), 2017.

[48] J. A. Tropp, "Freedman's inequality for matrix martingales," *Electron. Commun. Probab*, vol. 16, pp. 262–270, 2011.

[49] P.-Å. Wedin, "Perturbation bounds in connection with singular value decomposition," *BIT Numerical Mathematics*, vol. 12, no. 1, pp. 99–111, 1972.

[50] T. T. Cai and A. Zhang, "Rate-optimal perturbation bounds for singular subspaces with applications to high-dimensional statistics," *The Annals of Statistics*, vol. 46, no. 1, pp. 60–89, 2018.