Compressed Decentralized Learning of Conditional Mean Embedding Operators in Reproducing Kernel Hilbert Spaces

Boya Hou, Sina Sanjari, Nathan Dahlin, Subhonmesh Bose

Department of Electrical and Computer Engineering, University of Illinois Urbana-Champaign, Urbana, IL 61801.
{boyahou2,sanjari,dahlin,boses}@illinois.edu

Abstract

Conditional mean embedding (CME) operators encode conditional probability distributions within reproducing kernel Hilbert spaces (RKHS). In this paper, we present a decentralized algorithm for a collection of agents to cooperatively approximate CME over a network. Communication constraints limit the agents from sending all data to their neighbors; we only allow *sparse* representations of covariance operators to be exchanged among agents, compositions of which defines CME. Using a coherence-based compression scheme, we present a consensus-type algorithm that preserves the average of the approximations of the covariance operators across the network. We theoretically prove that the iterative dynamics in RKHS is stable. We then empirically study our algorithm to estimate CMEs to learn spectra of Koopman operators for Markovian dynamical systems and to execute approximate value iteration for Markov decision processes (MDPs).

1 Introduction

Over the last several years, embeddings of conditional probability distributions into reproducing kernel Hilbert spaces (RKHS), known as conditional mean embeddings (CMEs), have gained recognition as a powerful tool for both describing and estimating the dynamics of unknown or uncertain nonlinear dynamical systems. The CME framework captures transition dynamics without resorting to explicit modeling of system dynamics such as differential equations, yielding a representation which reduces computationally intensive, high dimensional integrations to inner products with linear complexity. Moreover, convergence guarantees on data-driven estimates of the embeddings are available, e.g., (Song et al. 2009; Grünewälder et al. 2012).

In many applications involving dynamical systems, a distributed, multi-agent approach to sensing and estimation is desired, if not required. The training time incurred in industrial-scale machine learning, for example, can be reduced when model estimation is decentralized across parallelized computing resources (Koppel et al. 2018). Operations involving systems which naturally encompass large geographical areas, such as wind condition assessment, necessitate deployment of multi-robot sensing teams in order to maintain feasible exploration time (Bobade, Panagou, and Kurdila 2019). In fully decentralized scenarios, individual agents or sensors share observations and other information across a communication network in order to reach a consensus regarding quantities of interest. However, this process is complicated by constraints arising from characteristics of the network and the agents themselves, including limitations on sensing, communication and computation.

In this work, our particular interest lies in a decentralized, networked multi-agent setting where agents cooperatively seek to learn CME operators describing a common dynamical system, while interacting across a network with constraints on the capacity of communication links. Specifically, agents communicate their tentative empirical estimates with neighbors in order to augment and refine their system model estimates. The empirical estimators maintained by each agent are represented using kernel functions centered at data points and weight coefficients locally observed and computed, whose size scales with the number of samples. Thus, with large amount of data, the required inter-agent exchanges may exceed network link capacities.

Scalability has long been identified as a critical issue facing kernel methods, e.g., see (Lever et al. 2016). As such, it has generated a considerable body of literature developing *sparsification* techniques that attempt to discard data found to be essentially redundant for the purposes of estimation (Engel, Mannor, and Meir 2002; Kivinen, Smola, and Williamson 2004; Richard, Bermudez, and Honeine 2008; Koppel et al. 2017; Hou, Bose, and Vaidya 2021). Here, we draw upon this work to design a compression protocol executed by agents prior to transmission. Specifically, we adopt the notion of *coherency* to control message length. To the best of our knowledge, this is the first work that provides detailed theoretical analysis on compressed decentralized learning in RKHS.

Our approach is most closely related to the scalar-valued quantized averaging techniques found in the networked consensus literature (Kashyap, Başar, and Srikant 2007; El Chamie, Liu, and Başar 2016), as the resulting evolution of estimates involves nonlinear dynamics, wherein the initial average across agents is not necessarily preserved (Nedic et al. 2009; Carli et al. 2010; El Chamie, Liu, and Başar 2016; Frasca et al. 2009; Aysal and Barner 2010). Nevertheless, our problem differs from the aforementioned ones in many respects. First, we consider learning operators rather than real values. Second, the data dependent, coherence-based compression mapping cannot be viewed as that generated by uniform quantizers studied in quantized consensus.

The main contribution of this paper is the design of a coherence-based compressed decentralized algorithm that aims to learn the CME operator. While decentralized learning over RKHS is well-studied, its sparse variant is much more challenging to address, as the errors due to approximations can pile up over time, making the estimate biased. We show in Theorem 1 that under mild conditions on the communication network, the initial average of empirical (cross) covariance operators across the network is preserved and each agent obtains a "stationary dictionary". In addition, we prove the stability of local empirical CME operators maintained by each agent through the iterative dynamics. Finally, we present applications of our consensus algorithm as a basis for distributed estimation of eigenfunctions of kernel transfer operators in a pair of oscillator systems, and approximate value iteration in a benchmark reinforcement learning task.

2 Preliminaries: RKHS and CME

We briefly review requisite definitions and properties of RKHS, and refer to (Muandet et al. 2016) for a detailed exposition. Let $\mathbb{X}$ be a compact subset of an appropriate dimensional Euclidean space, and $\kappa : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ be a symmetric, positive semi-definite kernel. Define $\mathcal{H}$ as the RKHS defined via the kernel κ as the completion of the linear span of $\{\phi(x) := \kappa(x, \cdot) : x \in \mathbb{X}\}$ that is equipped with the inner product $\langle \cdot, \cdot \rangle$, satisfying $\langle \phi(x), \phi(y) \rangle = \kappa(x, y)$. Here, ϕ is called the *feature map* of the kernel κ. The inner product satisfies the reproducing property, given by

$$\langle \phi(x), f \rangle = f(x), \quad \forall x \in \mathbb{X}, \quad f \in \mathcal{H}. \tag{1}$$

Kernel mean embedding of probability distributions Consider a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with a σ-algebra $\mathcal{F}$ and a probability measure $\mathbb{P}$. Let $X : (\Omega, \mathcal{F}, \mathbb{P}) \to (\mathbb{X}, \Sigma, \mathbb{P}_X)$ be a $\mathbb{X}$-valued random variable, where Σ is the Borel σ-algebra on $\mathbb{X}$ and $\mathbb{P}_X$ is a distribution on X. Denote $\mathbb{E}_X$ as the expectation with respect to $\mathbb{P}_X$. If κ is $\Sigma \otimes \Sigma$-measurable, and $\mathbb{E}_X \left[\sqrt{\kappa(x, x)} \right] < \infty$, then the probability distribution $\mathbb{P}_X$ can be embedded within the RKHS $\mathcal{H}$ (see e.g, (Muandet et al. 2016, Lemma 3.1)). In fact, there exists a *kernel mean embedding* $\mu : \mathbb{P}_X \mapsto \mu_{\mathbb{P}_X} \in \mathcal{H}$ such that

$$\mu_{\mathbb{P}_X} := \mathbb{E}_X \left[\kappa(X, \cdot) \right]. \tag{2}$$

Throughout this paper, we suppose that κ is measurable and bounded, i.e., $\sup_{x \in \mathbb{X}} \kappa(x, x) < \infty$.

Let $Y : (\Omega_2, \mathcal{F}_2, \mathbb{P}_2) \to (\mathbb{Y}, \Sigma_2, \mathbb{P}_Y)$ be a $\mathbb{Y}$-valued random variable. In addition, let $(\mathcal{H}_1, \kappa_1), (\mathcal{H}_2, \kappa_2)$ be RKHSs on $\mathbb{X}$ and $\mathbb{Y}$, respectively. Denote $\mathbb{P}_{XY}$ as the joint distribution of (X, Y), and $\mathbb{E}_{XY}$ as the expectation with respect to $\mathbb{P}_{XY}$. Along the same lines as the preceding discussion, per (Berlinet and Thomas-Agnan 2011), $\mathbb{P}_{XY}$ can be embedded into the tensor product Hilbert space $\mathcal{H}_1 \otimes \mathcal{H}_2$ as

$$\mu_{\mathbb{P}_{XY}} := \mathcal{C}_{XY} = \mathbb{E}_{XY}[\phi(X) \otimes \psi(Y)], \tag{3}$$

where $\mathcal{C}_{XY}$ is the (uncentered) *cross-covariance* operator and ψ is the feature map of $\mathcal{H}_2$. This operator can be identified as an element in $\mathcal{H}_1 \otimes \mathcal{H}_2$ with kernel $\kappa_\otimes := \kappa_1 \otimes \kappa_2$, i.e.,

$$\kappa_\otimes \bigg((x_1, y_1), (x_2, y_2) \bigg) = \kappa_1(x_1, x_2)\, \kappa_2(y_1, y_2), \tag{4}$$

for all $x_1, x_2 \in \mathbb{X}$, $y_1, y_2 \in \mathbb{Y}$ and (joint) feature map

$$\varphi\left(x_i, y_i\right) := \phi\left(x_i\right) \otimes \psi\left(y_i\right) = \kappa_1\left(x_i, \cdot\right) \kappa_2\left(y_i, \cdot\right). \tag{5}$$

Equivalently, we can also view $\mathcal{C}_{XY}$ as a Hilbert-Schmidt (HS) operator $\mathcal{C}_{XY} : \mathcal{H}_2 \to \mathcal{H}_1$ that satisfies

$$\mathbb{E}_{XY}[f(X)g(Y)] = \langle \mathcal{C}_{XY} g, f \rangle, \quad \forall f \in \mathcal{H}_1, g \in \mathcal{H}_2. \tag{6}$$

Along the same lines as above, the *(uncentered) covariance* operator $\mathcal{C}_{XX}$, embedding of the marginal distribution $\mathbb{P}_X$ into $\mathcal{H}_1 \otimes \mathcal{H}_1$, can be defined as

$$\mathcal{C}_{XX} := \mathbb{E}_X[\phi(X) \otimes \phi(X)]. \tag{7}$$

One often needs to estimate such mean embeddings from data. Let $\mathcal{D} := \{(x_1, y_1), \ldots, (x_n, y_n)\}$ denote a collection of n data points sampled independently according to $\mathbb{P}_{XY}$. The empirical estimations of $\mathcal{C}_{XX}$ and $\mathcal{C}_{XY}$, given $\mathcal{D}$, then are

$$\begin{aligned} C_{XX} &= \frac{1}{n} \sum_{i=1}^{n} \phi\left(x_i\right) \otimes \phi\left(x_i\right) = \frac{1}{n} \sum_{i=1}^{n} \varphi\left(x_i, x_i\right), \\ C_{XY} &= \frac{1}{n} \sum_{i=1}^{n} \phi\left(x_i\right) \otimes \psi\left(y_i\right) = \frac{1}{n} \sum_{i=1}^{n} \varphi\left(x_i, y_i\right). \end{aligned} \tag{8}$$

The empirical estimation in (8) is known be $\sqrt{n}$-consistent in RKHS norm (Muandet et al. 2016; Tolstikhin, Sriperumbudur, and Muandet 2017; Hou, Bose, and Vaidya 2021).

Embedding of conditional probability distributions Let $\mathbb{P}_{Y|x}$ denote the conditional distribution of the random variable Y given $X = x \in \mathbb{X}$. The embedding of $\mathbb{P}_{Y|x}$ into $\mathcal{H}_2$ is defined as

$$\mu_{\mathbb{P}_{Y|x}} := \mathbb{E}_{Y|x}[\phi(Y)|X = x] \qquad \forall x \in \mathbb{X}. \tag{9}$$

In fact, per (Song et al. 2009), the conditional mean embedding operator $\mathcal{U}_{Y|X} : \mathcal{H}_1 \to \mathcal{H}_2$, is a linear operator that satisfies

$$\mu_{\mathbb{P}_{Y|x}} = \mathcal{U}_{Y|X} \phi(x), . \tag{10}$$

In particular, $\mathcal{U}_{Y|X} : \mu_{\mathbb{P}_X} \mapsto \mu_{\mathbb{P}_Y}$, i.e.,

$$\mu_{\mathbb{P}_Y} = \mathcal{U}_{Y|X} \mu_{\mathbb{P}_X}. \tag{11}$$

If $\mathbb{E}_{Y|x}[f(Y)|X = x] \in \mathcal{H}_2$ for all $f \in \mathcal{H}_2$ and $x \in \mathbb{X}$, and $\mathcal{C}_{XX}$ is invertible, then we have

$$\mathcal{U}_{Y|X} = \mathcal{C}_{YX} \mathcal{C}_{XX}^{-1}. \tag{12}$$

The operator definitions in the next section rely on $\mathcal{C}_{XX}$ being an invertible map. For technical reasons, (Song et al. 2009) we consider their regularized versions, defined as

$$\mathcal{U}_\varepsilon = \mathcal{C}_{YX} \left(\mathcal{C}_{XX} + \varepsilon \mathcal{I}\right)^{-1}, \tag{13}$$

for $\varepsilon > 0$, where $\mathcal{I}$ is the identity operator. Given the collection of data points $\mathcal{D} := \{(x_1, y_1), \ldots, (x_n, y_n)\}$, the regularized empirical estimate of the CME operator becomes

$$U_\varepsilon := C_{YX} \left(C_{XX} + \varepsilon I\right)^{-1}. \tag{14}$$

3 Applications of CME in Markov Processes

CMEs encode how the distribution of one random variable relates to another's. If the random variables correspond to successive states of a discrete-time Markov process, CMEs naturally encapsulate the transition dynamics of that process. There are two main reasons that make CME particularly suitable to study Markovian dynamics. First, the approximation errors in CME estimation is independent of the dimension of the state space of the underlying data (Song et al. 2009, Theorem 6). Second, evaluations of high-dimensional integration, such as those required to evaluate expectations over the state space, reduce to the simple computation of an inner product in an RKHS. Here, we briefly review how CME is related to transfer operators and the control of MDPs.

3.1 Uncontrolled Markov Processes

Consider a time-homogeneous discrete-time Markov process over $\mathbb{X}$, described by the transition kernel density p as

$$\mathbb{P}\left\{X_{t+1} \in A \mid x_t = x\right\} = \int_A p(y|x)\mathrm{d}y, \qquad \forall t \geq 0 \quad (15)$$

for any Borel set A in $\mathbb{X}$, where the random variable X_t denotes the state at time t. If f describes a probability density of states in $\mathbb{X}$, then the *Perron–Frobenius operator* $\mathcal{P}$, propagates f through the dynamical system, i.e.,

$$\left(\mathcal{P}f\right)(y) = \int p(y|x)f(x)\mathrm{d}x \qquad \forall y \in \mathbb{X}. \quad (16)$$

On the other hand, for any function f on $\mathbb{X}$, the *Koopman operator* $\mathcal{K}$ acts on f such that

$$\left(\mathcal{K}f\right)(x) = \int p(y|x)f(y)\mathrm{d}y \qquad \forall x \in \mathbb{X}. \quad (17)$$

Using these operators, the nonlinear probabilistic propagation of a finite-dimensional state can be described via the linear propagation of infinite-dimensional densities or functions of states. Comparing (11) and (16) motivates the definition of Perron–Frobenius operator as $\mathcal{P} := \mathcal{U}_{Y|X}$ (see (Klus, Schuster, and Muandet 2020) for details). As a result, we have

$$\mathcal{P} = \mathcal{U}_{Y|X} = \mathcal{C}_{YX}\mathcal{C}_{XX}^{-1}. \quad (18)$$

Along the same line as (18), we identify the Koopman operator $\mathcal{K}$ as the adjoint of $\mathcal{U}_{Y|X} = \mathcal{P}$, given by

$$\mathcal{K} = \mathcal{C}_{XX}^{-1}\mathcal{C}_{XY}. \quad (19)$$

The spectra of these linear transfer operators contain valuable information about the dynamical system, including their regions of attraction and stable orbits (Mezić 2005). In view of (18) and (19), the interactions of these operators with RKHS can be studied through the CME operator. Data-driven approximations to CME, e.g., in (Song et al. 2009), then allow data-driven dynamical systems analysis.

3.2 Markov Decision Processes (MDPs)

CMEs also find application in the analysis of MDPs. Consider an MDP with compact state and action spaces $\mathbb{X}$ and $\mathbb{U}$ that are subsets of a finite-dimensional Euclidean space. The state dynamics are described by a transition kernel $x_{t+1} \sim p(\cdot|x_t, u_t)$. The value function at $x \in \mathbb{X}$ (expected cost starting from x) satisfies

$$(\mathcal{B}V)(x) := \min_{u \in \mathbb{U}} \left\{c(x,u) + \gamma \mathbb{E}_{Y|(x,u)}[V(Y)]\right\}, \quad (20)$$

where in this context, Y is the $\mathbb{X}$-valued random state following x, and $c : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ is the immediate cost function. Given an arbitrary V_0, the sequence $\{V_k\}$ defined via $V_{k+1} = \mathcal{B}V_k$ converges in sup-norm to an optimal value function (Szepesvári 2010).

CMEs allow for "kernelization" of the value iteration procedure. Let $\mathbb{Z} = \mathbb{X} \times \mathbb{U}$ and Z be a $\mathbb{Z}$ valued random variable, and $(\mathcal{H}_1, \kappa_1)$ and $(\mathcal{H}_2, \kappa_2)$ be RKHS on $\mathbb{Z}$ and $\mathbb{X}$, respectively. For $f \in \mathcal{H}_2$, the mapping $f \mapsto \mathbb{E}_{Y|z}[f(Y)]$ can be implemented using a CME as

$$\mathbb{E}_{Y|z}[f(Y)] = \langle f, \mu_{Y|z}\rangle, \quad (21)$$

where $\mu_{Y|z}$ is the distribution on Y, conditioned on current state-action pair $z = (x, u)$. Then, $\mu_{Y|z} \in \mathcal{H}_2$ is given by

$$\mu_{Y|z} = \mathcal{C}_{YZ}\mathcal{C}_{ZZ}^{-1}\phi(z), \quad (22)$$

where ϕ denotes the feature map corresponding to κ_2 (Song et al. 2009; Grunewalder et al. 2012).

In the next section, we propose a distributed (decentralized) algorithm such that when the transition kernel is unknown in these Markov processes, the empirical approximations of (19) and (21) can be learned by agents in a network, having access only to private local data.

4 Compressed Decentralized Learning of the CME Operator

Consider a finite collection of agents $\mathcal{V} := \{1, \ldots, m\}$ that are connected via a communication network. The agents seek to collectively learn an empirical approximation of CME. A single agent can utilize (8) and (14) to learn an approximation of the CME operator from data. Simulation capabilities can limit the amount of data that a single agent can collect. Thus, our goal is to allow agents to share their approximations with each other to create meaningful approximations of the CME operator. Represent the network by a fixed undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the nodes (agents) and the edges (communication channels) of the graph. Agents can exchange their local tentative empirical approximations of the CME operator at time $t \geq 0$ with their neighbors in $\mathcal{G}$. Assume that there is no communication delay in the network.

Let $\mathcal{D}^i(0)$ denote the local data points available to agent i at $t = 0$. The CME operator approximations are obtained from the covariance operator approximations in (7). These approximations require the knowledge of each of the data points. Since the number of data points in $\mathcal{D}^i(0)$ can be large, broadcasting that list over the network can be overwhelming. The difficulty of handling large data in kernel estimation is well-documented, e.g., see Engel, Mannor, and Meir (2002); Koppel et al. (2017, 2018); Forero, Cano, and Giannakis (2010). To circumvent this difficulty, we allow each agent $i \in \mathcal{V}$ to share only *compressed* empirical approximations of the CME operators with their neighbors. This local compressed approximations are constructed by pruning the local dataset $\mathcal{D}^i(t)$ for each agent i. Our notion of compression is based on *coherency*; see Richard, Bermudez, and Honeine (2008). Under this compression technique, only data points that are not "too similar" are retained, where similarity is measured via normalized inner products between the two points, induced by the kernel functions. In what follows, we define coherency and present our algorithm for decentralized CME estimation.

Let $\mathcal{D}^i(t)$ denote the local dataset available to agent i at time t with cardinality $D^i(t) := |\mathcal{D}^i(t)|$. Let κ be a positive definite reproducing kernel on $\mathcal{D}^i(t)$ with induced RKHS $\mathcal{H}_i(t)$. Given $\mathcal{D}^i(t)$ and γ, each agent constructs a γ-coherent compressed dataset $\widehat{\mathcal{D}}^i(t) \subseteq \mathcal{D}^i(t)$, by identifying a subset that satisfies

$$|\kappa(x_s^\star, x_\tau^\star)| \leq \sqrt{\gamma \kappa(x_s^\star, x_s^\star)\kappa(x_\tau^\star, x_\tau^\star)}, \tag{23}$$

for each s, τ where $(x_s^\star, x_\tau^\star)$ is either (x_s, x_τ) or (x_s^+, x_τ^+), and $(x_s, x_s^+), (x_\tau, x_\tau^+)$ are in $\mathcal{D}_\gamma$. For any γ, we denote the compression operation by

$$\text{CMP}_\gamma(\mathcal{D}^i(t)) = \widehat{\mathcal{D}}^i(t). \tag{24}$$

Let $W \in \mathbb{R}^{m \times m}$ be a weighted adjacency matrix of the graph $\mathcal{G}$. Thus, we have $w_{i,j} > 0 \iff (i,j) \in \mathcal{E}$. Let $\mathcal{N}^i$ denote the neighbors of i in $\mathcal{G}$. We design an algorithm for agents to arrive at empirical covariance estimations $C^i_{YX}(t)$ and $C^i_{XX}(t)$ using compressed variants from neighbors over time, to then calculate the CME operator

$$U^i_\varepsilon(t) := C^i_{YX}(t)\left(C^i_{XX}(t) + \varepsilon I\right)^{-1}, \quad \varepsilon > 0. \tag{25}$$

Our decentralized algorithm proceeds as follows. At $t = 0$, given local dateset $\mathcal{D}^i(0)$, let κ be a reproducing kernel on $\mathcal{D}^i(0)$ with RKHS $\mathcal{H}_i(0)$. Each agent i constructs its local estimate $C^i_{YX}(0)$ and $C^i_{XX}(0)$ via (8). Then, each agent compresses the local data set $\mathcal{D}^i(0)$ with the γ-coherent compression scheme CMP_γ to obtain the compressed dataset $\widehat{\mathcal{D}}^i(0)$. Let $\mathcal{I}^i(0) = \{1, \cdots, D^i(0)\}$ be the index set for $\mathcal{D}^i(0)$ and $\widehat{\mathcal{I}}^i(0)$ be a subset of $\mathcal{I}^i(0)$, containing all $k \in \mathcal{I}^i(0)$ for which the pair (x_k, y_k) is in $\widehat{\mathcal{D}}^i(0)$. The sparse estimators of $C^i_{XY}(0)$ and $C^i_{XX}(0)$ for agent i are given by $\widehat{C}^i_{YX}(0) := \sum_{k\in\widehat{\mathcal{I}}^i(0)} \widehat{\alpha}^i_k(0)\varphi\left(x_k, y_k\right)$, $\widehat{C}^i_{XX}(0) := \sum_{k\in\widehat{\mathcal{I}}^i(0)} \widehat{\beta}^i_k(0)\varphi\left(x_k, x_k\right)$, where real-valued vector $\widehat{\boldsymbol{\alpha}}^i(0) := (\widehat{\alpha}^i_k(0))_{k\in\widehat{\mathcal{I}}^i(0)}$ (and similarly, $\widehat{\boldsymbol{\beta}}^i(0) := (\widehat{\beta}^i_k(0))_{k\in\widehat{\mathcal{I}}^i(0)}$) is defined as[1]

$$\begin{aligned}\widehat{\boldsymbol{\alpha}}^i(0) := \underset{\overline{\boldsymbol{\alpha}}(0)}{\operatorname{argmin}} \Bigg\| &\frac{1}{D^i(0)} \sum_{k\in\mathcal{I}^i(0)} \varphi\left(x_k, y_k\right) \\ &- \sum_{k\in\widehat{\mathcal{I}}^i(0)} \overline{\alpha}_k(0)\varphi\left(x_k, y_k\right)\Bigg\|^2_{\mathcal{H}_i(0)},\end{aligned} \tag{26}$$

so that the contribution of discarded kernel function can be distributed over elements in $\widehat{\mathcal{D}}^i(0)$. At $t = 1$, agent i communicates $\widehat{\mathcal{D}}^i(0)$, $\widehat{C}^i_{YX}(0)$ and $\widehat{C}^i_{XX}(0)$ to each of its neighbor $j \in \mathcal{N}^i$. Now, each agent must update its local estimate of the covariance operators using a combination of its own prior estimate and the compressed estimates that it receives from its neighbors.

A natural candidate for consensus-based update rule for covariance operator estimates takes the form

$$\widehat{C}^i_{YX}(t+1) = w_{i,i}\widehat{C}^i_{YX}(t) + \sum_{j\in\mathcal{N}^i} w_{i,j}\widehat{C}^j_{YX}(t) \tag{27}$$

for $t \geq 0$. However, these updates may cause the average across the network to deviate from its initial value.

Instead, we consider a modified update rule for $C^i_{YX}(t)$ (similarly for $C^i_{XX}(t)$), given by

$$\begin{aligned}C^i_{YX}(t+1) =& w_{i,i}\widehat{C}^i_{YX}(t) + \sum_{j\in\mathcal{N}^i} w_{i,j}\widehat{C}^j_{YX}(t) \\ &+ \underbrace{C^i_{YX}(t) - \widehat{C}^i_{YX}(t)}_{e^i_{YX}(t)}.\end{aligned} \tag{28}$$

Our method is formally presented in Algorithm 1.

The proposed algorithm proceeds in two stage – dictionary passing and coefficient passing. Let $\overline{d(\mathcal{G})}$ denote the longest path between any two nodes of graph $\mathcal{G}$. For $t = 0, \ldots, \overline{d(\mathcal{G})}$, each agent i sends its compressed dictionary $\widehat{\mathcal{D}}^i(t)$ to its neighbors in $\mathcal{G}$. From $t = \overline{d(\mathcal{G})}$ onwards, agents exchange their $\widehat{\mathcal{D}}^i(t)$ and coefficient vectors $\boldsymbol{\alpha}^i(t), \boldsymbol{\beta}^i(t)$. As we will prove in the next section, agents arrive at a stationary dictionary within $\overline{d(\mathcal{G})}$ time-steps. Then, CME learning with compressed dictionaries leads to a modified consensus dynamics described by (28).

Dictionary passing: Agent i begins by creating an ordering for its neighbors and appends itself to the start of this list, i.e., it generates the ordered list $\text{ORD}^i = \{v^i_1, \cdots, v^i_{|\mathcal{N}^i|+1}\}$ from $\{i\}\bigcup\mathcal{N}^i$ with $v^i_1 = i$. With the local dictionary $\mathcal{D}^i(t)$ at time t, agent i prunes it to obtain a γ-coherent dictionary $\widehat{\mathcal{D}}^i(t) = \text{CMP}_\gamma\big(\mathcal{D}^i(t)\big)$. Then, it shares its compressed dictionary $\widehat{\mathcal{D}}^i(t)$ with its neighbors. Upon receiving the dictionaries from its neighbors, agent i updates $\mathcal{D}^i(t)$ as

$$\mathcal{D}^i(t) = \mathcal{D}^i(t-1)\bigcup\left(\bigcup_{j\in\mathcal{N}^i}\widehat{\mathcal{D}}^j(t-1)\right). \tag{29}$$

For our analysis, we compress $\mathcal{D}^i(t)$ to obtain $\widehat{\mathcal{D}}^i(t)$ in a specific way. We first construct a temporary collection $\widetilde{\mathcal{D}}^i(t)$ via concatenation of received compressed dictionaries as

$$\widetilde{\mathcal{D}}^i(t) := \text{CONCAT}\Big[\widehat{\mathcal{D}}_{v^i_1}(t-1), \ldots, \widehat{\mathcal{D}}_{v^i_{|\mathcal{N}^i|+1}}(t-1)\Big].$$

It then computes the Gram matrix using all elements in $\widetilde{\mathcal{D}}^i$ with $\kappa_\otimes$ as the kernel. If two data points (x_s, y_s) and (x_t, y_t) are such that (23) fails to hold with indices $s < t$, agent

[1] $\widehat{C}^i_{YX}(0)$ is the orthogonal projection of $C^i_{XY}(0)$ onto the closed subspace $\text{span}\{\varphi(x_k, y_k) : k \in \widehat{\mathcal{I}}^i_s\} \subseteq \mathcal{H}_\otimes$.

i retains (x_s, y_s) in $\widehat{\mathcal{D}}^i$, but discards (x_t, y_t) from $\widehat{\mathcal{D}}^i$ and correspondingly removes the row/column associated with (x_t, y_t) from the Gram matrix. It repeats this operation until all elements in the Gram matrix satisfies (23) to obtain

$$\widehat{\mathcal{D}}^i(t) = \text{CMP}_\gamma\left(\widetilde{\mathcal{D}}^i(t)\right). \tag{30}$$

Coefficient passing: We show in Theorem 1 (b) that the dictionaries of all agents stabilize by $t = \overline{d(\mathcal{G})}$. Denote the stationary dictionary of agent i and its compression as $\mathcal{D}_s^i$ and $\widehat{\mathcal{D}}_s^i$, respectively, for each $i \in \mathcal{V}$. After $t = \overline{d(\mathcal{G})}$, the agents only update coefficients they assign to the elements in their stationary dictionaries to obtain their CME estimates.

Let $\mathcal{I}_s^i = \{1, \cdots, D_s^i\}$ be the index set for $\mathcal{D}_s^i$ and $\widehat{\mathcal{I}}_s^i$ be a subset of $\mathcal{I}_s^i$, containing all $k \in \mathcal{I}_s^i$ for which the pair (x_k, y_k) is in $\widehat{\mathcal{D}}_s^i$. At $t = \overline{d(\mathcal{G})}$, each agent i forms estimates $C_{YX}^i(t)$ (likewise $C_{XX}^i(t)$) by assigning uniform weights to each element in $\mathcal{D}^i(0)$ and 0 to elements in $\mathcal{D}_s^i \setminus \mathcal{D}^i(0)$ as

$$\begin{aligned} C_{YX}^i(\overline{d(\mathcal{G})}) &= \sum_{k \in \mathcal{I}_s^i} \alpha_k^i(\overline{d(\mathcal{G})}) \varphi\left(x_k, y_k\right), \\ \alpha_k^i(\overline{d(\mathcal{G})}) &= \begin{cases} \frac{1}{D^i(0)}, & \text{if } k \in \mathcal{I}^i(0), \\ 0, & \text{if } k \in \mathcal{I}_s^i \setminus \mathcal{I}^i(0). \end{cases} \end{aligned} \tag{31}$$

Each agent i then computes a CME estimate via (25). The compressed counterpart of $\widehat{C}_{YX}^i(t)$ is given by

$$\widehat{C}_{YX}^i(t) := \sum_{k \in \widehat{\mathcal{I}}_s^i} \widehat{\alpha}_k^i(t) \varphi\left(x_k, y_k\right) \tag{32}$$

with $\widehat{\boldsymbol{\alpha}}^i(t) := (\widehat{\alpha}_k^i(t))_{k \in \widehat{\mathcal{I}}_s^i}$ that solves

$$\begin{aligned} &\underset{\boldsymbol{z}}{\text{argmin}} \left\| \sum_{k \in \widehat{\mathcal{I}}_s^i} z_k \varphi\left(x_k, y_k\right) - \sum_{k \in \mathcal{I}_s^i} \alpha_k^i(t) \varphi\left(x_k, y_k\right) \right\|_{\mathcal{H}_i}^2 \\ &= [G_{\widehat{\mathcal{D}}^i, \widehat{\mathcal{D}}^i}]^{-1} G_{\widehat{\mathcal{D}}^i, \mathcal{D}^i} \boldsymbol{\alpha}^i(t), \end{aligned} \tag{33}$$

where $G_{\widehat{\mathcal{D}}^i, \mathcal{D}^i}$ is the Gram matrix whose (p, q)-th entry is $\kappa_\otimes\big((x_p, y_p), (x_q, y_q)\big)$ for (x_p, y_p) in $\widehat{\mathcal{D}}_s^i$ and (x_q, y_q) in $\mathcal{D}_s^i$. $G_{\widehat{\mathcal{D}}^i, \widehat{\mathcal{D}}^i}$ is defined similarly. Notice that $\widehat{\mathcal{D}}_s^i$ is a γ-coherent dictionary. From Gershgorin's disk theorem (Richard, Bermudez, and Honeine 2008), it follows that the smallest eigenvalue of $G_{\widehat{\mathcal{D}}^i, \widehat{\mathcal{D}}^i}$ is bounded away from zero. Therefore, $G_{\widehat{\mathcal{D}}^i, \widehat{\mathcal{D}}^i}$ is invertible and $\widehat{\boldsymbol{\alpha}}$ is well-defined for every choice of $\gamma \in (0, 1)$. One can similarly compute the compressed counterpart $\widehat{C}_{XX}^i(t)$ of $C_{XX}^i(t)$ using $\widehat{\boldsymbol{\beta}}^i(t) := (\widehat{\beta}_k^i(t))_{k \in \widehat{\mathcal{I}}_s^i}$.

At time $t + 1$, agent i communicates its compressed estimates $\widehat{C}_{YX}^i(t)$ and $\widehat{C}_{XX}^i(t)$ to its neighbors. Then, it updates its estimate of $C_{YX}^i(t + 1)$ and $C_{XX}^i(t + 1)$, along with $\boldsymbol{\alpha}^i(t + 1)$and $\boldsymbol{\beta}^i(t + 1)$ using (28). When computing compressed estimators $\widehat{C}_{YX}^i(t + 1)$ and $\widehat{C}_{XX}^i(t + 1)$ via (32), the associated weights $\widehat{\boldsymbol{\alpha}}^i(t + 1)$ and $\widehat{\boldsymbol{\beta}}^i(t + 1)$ are updated according to (33).

Algorithm 1: Compressed decentralized learning of CME by agent i for all $i \in \mathcal{V}$

```
Require: Initial datasets $\mathcal{D}^i(0)$; Kernels $\kappa_1, \kappa_2$; Coherent parameter $\gamma$; Weighted adjacency matrix $W$; Longest path between any two agents $\overline{d(\mathcal{G})}$; Local ordering ORD$^i$; Time $t = 0$.
 1: Initialize $\widehat{\mathcal{D}}^i(0) \leftarrow \text{CMP}_\gamma\big(\mathcal{D}^i(0)\big)$
 2: for $t = 1, \cdots, \overline{d(\mathcal{G})}$ do
 3:     Update $\mathcal{D}^i(t)$ and $\widehat{\mathcal{D}}^i(t)$ via (29) and (30)
 4: end for
 5: Set $t = \overline{d(\mathcal{G})}$. Initialize $C_{YX}^i(t)$, $C_{XX}^i(t)$ as (31)
 6: for $t > \overline{d(\mathcal{G})}$ do
 7:     Send $\widehat{\mathcal{D}}^i(t-1), \widehat{\boldsymbol{\alpha}}^i(t-1), \widehat{\boldsymbol{\beta}}^i(t-1)$ to $j \in \mathcal{N}^i, i \in \mathcal{V}$
 8:     Update $C_{YX}^i(t)$, $C_{XX}^i(t)$ via (28)
 9:     Update $U_\varepsilon^i(t)$ via (25)
10:     Update $\widehat{C}_{YX}^i(t)$, $\widehat{C}_{XX}^i(t)$ via (32) and (33)
11: end for
```

5 Theoretical Analysis

We now present important properties of our decentralized CME operator estimation algorithm.

Theorem 1. *Suppose W is aperiodic, irreducible, and doubly stochastic and κ_1, κ_2 are bounded, continuous kernel functions. Then, updates by all agents according to Algorithm 1 satisfy the following assertions.*

(a) $\frac{1}{m}\sum_{i=1}^m C_\star^i(t) = \frac{1}{m}\sum_{i=1}^m C_\star^i(0)$ *for all* $t \geq 0$ *and* $\star \in \{XY, XX\}$.

(b) $\mathcal{D}^i(t)$ *remains the same as* $\mathcal{D}^i(\overline{d(\mathcal{G})})$, *even if line 3 is executed for all* $t \geq \overline{d(\mathcal{G})}$.

(c) For any $\delta > 0$, *there exists* $\epsilon > 0$, *such that*

$$\sup_i \|C_\star^i(0)\|_{\mathcal{H}_i} \leq \delta \implies \sup_i \|C_\star^i(t)\|_{\mathcal{H}_i} \leq \epsilon \tag{34}$$

for all $t \geq 0$ *and* $\star \in \{XY, XX\}$.

Our first result establishes the invariance of the network average of the covariance estimates. The second result shows that the dictionary passing in Algorithm 1 terminates within $\overline{d(\mathcal{G})}$ time to a stationary dictionary. The final result presents stability analysis for the CME operator under the compressed decentralized learning algorithm. The first two proofs are elementary. The third proof relies on analyzing the nonlinear dynamics of the coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ that define $C_\star^i(t)$. We rewrite this dynamics where the error due to compression is represented as a bounded input to a linear dynamical system. Its bounded input bounded output (BIBO) stability then allows us to infer the stability of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

Our result does not analyze asymptotic dynamics of the algorithm. We now present a stylized example next to illustrate that asymptotic dynamics of decentralized learning with approximations can be richer than convergence to an equilibrium point. To illustrate, equip $\mathbb{R}^n$ with inner product $\langle u, v\rangle := v^\top u$. Then, the kernel of $\mathbb{R}^n$ is the identity matrix (Manton and Amblard 2015, Definition 2.1). For simplicity and clarity, instead of working with covariance operators in the tensor product space, we work with empirical kernel mean embedding in such an RKHS.

Example 1. *Consider $m = 2$ agents connected via a network, whose initial dictionaries and their compressed variants be*

$$\mathcal{D}^i(0) = \{v, v^i\},\ \widehat{\mathcal{D}}^i(0) = \{v\}, \quad v, v^i \in \mathbb{R}^n \qquad (35)$$

for $i = 1, 2$. Assume for simplicity that $\|v\|_2^2 = 1$. At time $t = 0$, let the local empirical kernel mean embedding evaluated at local data points be given by $x^i(0) = {}^1\!/\!_2(v + v^i)$ for $i = 1, 2$. At time t, agent i's estimate becomes $\alpha_1^i(t)v + \alpha_2^i(t)v^i$. Then, the compressed variant is given by $\widehat{\alpha}^i(t)v$, where $\widehat{\alpha}^i(t) := x^i(t)^\top v$. Denote $\delta^{21}(0) := x^2(0) - x^1(0)$ and $c_\delta := \delta^{21}(0)^\top v$. Consider $W = [w, 1-w; 1-w, w]$, where $w \in [0, 1]$. Then, it can be shown that the estimates at time t are given by

$$\begin{aligned} x^1(t) &= x^1(0) - {}^1\!/\!_2((2w-1)^t - 1)c_\delta v, \\ x^2(t) &= x^2(0) + {}^1\!/\!_2((2w-1)^t - 1)c_\delta v. \end{aligned} \qquad (36)$$

For $w \in (0, 1)$, agents converge to compressed consensus

$$\lim_{t\to\infty} x^1(t) = x^1(0) + {}^1\!/\!_2 c_\delta v,\ \lim_{t\to\infty} x^2(t) = x^2(0) - {}^1\!/\!_2 c_\delta v.$$

With $w = 0$, we have

$$\begin{aligned} x^1(t) &= x^1(0) - {}^1\!/\!_2((-1)^t - 1)c_\delta v, \\ x^2(t) &= x^2(0) + {}^1\!/\!_2((-1)^t - 1)c_\delta v, \end{aligned}$$

which implies that $x^1(t)$ and $x^2(t)$ enter a limit cycle.

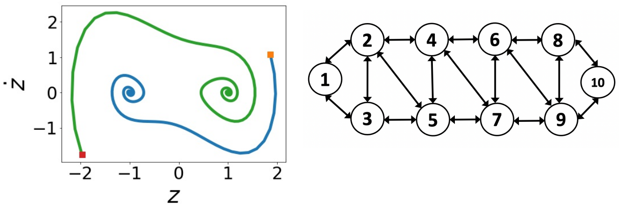
6 Numerical Experiments on Decentralized Learning of Kernel Transfer Operators

The spectra of transfer operators for dynamical systems are rich in information in that they can be used to study a variety of system characteristics including dominant modes, regions of attraction, stable orbits, etc. In this section, we report results on learning eigenfunctions of kernel transfer operators that can be estimated cooperatively via sparse, decentralized data exchange. See (Klus, Schuster, and Muandet 2020) for details on the procedure to construct eigenfunctions using Gram matrices alone.

Consider the unforced Duffing oscillator, described by

$$\ddot{z} = -\delta\dot{z} - z\left(\beta + \alpha z^2\right),$$

with $\delta = 0.5$, $\beta = -1$, and $\alpha = 1$, where $z \in \mathbb{R}$ and $\dot{z} \in \mathbb{R}$ are the scalar position and velocity, respectively. Let $x = (z, \dot{z})$. As Figure 1a reveals, this system exhibits two regions of attraction, corresponding to equilibrium points $x = (-1, 0)$ and $x = (1, 0)$.



(a) (b)

Figure 1: (a) Two trajectories of the Duffing oscillator that converge to two different equilibrium points. (b) Illustration of a network with 10 agents.

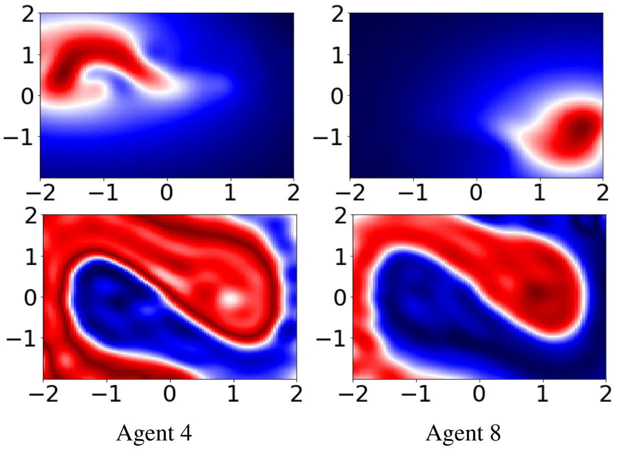


Figure 2: Leading eigenfunctions of $K_\varepsilon^i(t)$(the adjoint of $U_\varepsilon^i(t)$ at times $t = 0$ (top row) and $t = 17$ (bottom row) for $i = 4, 8$.

We consider a 10-agent network shown in Figure 1b. We choose Metropolis edge weights (Xiao, Boyd, and Lall 2005)

$$w_{i,j} = \frac{1}{1 + \max\{N^i, N^j\}}\ \forall\,(i,j) \in \mathcal{E},\ w_{i,i} = 1 - \sum_{j \in \mathcal{N}^i} w_{i,j},$$

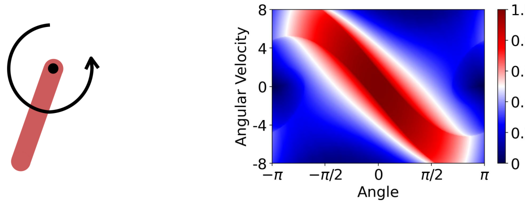
where $N^i = |\mathcal{N}^i|$. Each agent uses a combination of three Gaussian kernels: $\kappa(x_1, x_2) = \sum_{\ell=1}^{3} \eta_\ell \exp(-\|x_1 - x_2\|_2^2/(2\sigma_\ell^2))$, where $(\eta_1, \eta_2, \eta_3) = (0.5, 0.3, 0.2)$ and $(\sigma_1, \sigma_2, \sigma_3) = (1.45, 0.48, 0.29)$.

Initial local datasets are formed by partitioning 4900 data points uniformly sampled over $[z, \dot{z}] \in [-2, 2] \times [-2, 2]$ with $\Delta t = 0.25$ into ten sets of size 490 each. Heat-maps of the leading eigenfunction of the Koopman operator as estimated with initial local datasets by agents 4 and 8 are shown in the first row of Figure 2. We apply Algorithm 1 with $\gamma = 0.94$. Since $\overline{d(\mathcal{G})} = 9$, every agent obtains a stationary (compressed) dictionary at time $t = 9$ with the maximum cardinality of $\mathcal{D}_i$ for all $i \in \mathcal{V}$ being 1253 and that of $\widehat{\mathcal{D}}_i$ being 693. Once the dictionaries stabilize, agents continue to exchange information in order to reach consensus in their operator estimates. Second row in Figure 2 reveals that each agent detects the distinct regions of attraction.

7 Numerical Experiments on Decentralized Learning for Value Iteration

In this section, we apply the CME operator to perform approximate value iteration in a decentralized fashion.

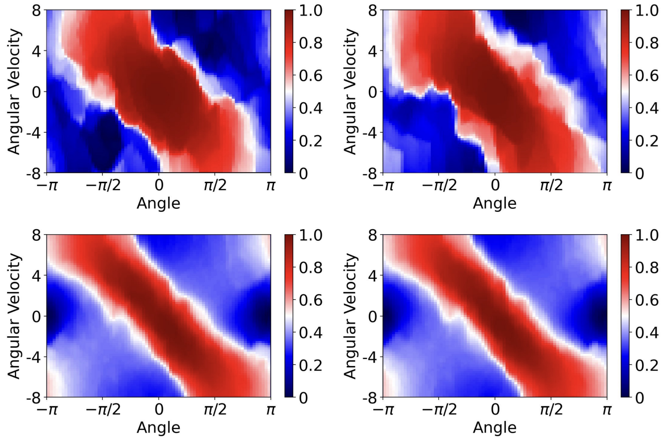
We consider a discrete-time approximation of the continuous-time frictionless pendulum dynamics as implemented in the OpenAI Gym package (Brockman et al. 2016). The approximated continuous system is governed by $\ddot{\theta}(t) = (3g/2l)\sin\theta(t) + (3/ml^2)u(t)$, where θ is the pendulum angle, g is the gravitational constant, $l = 1$m is the pendulum



(a)

(b)

Figure 3: (a) Pendulum swingup environment. (b) Reference value function using discretized state and action space.



Agent 2

Agent 10

Figure 4: Agent value functions for the inverted pendulum problem at time $t = 0$ (top row) and $t = 14$ (bottom row) with $\max_{i \in \mathcal{V}} |\mathcal{D}_i(14)| = 5296$, $\max_{i \in \mathcal{V}} |\widehat{\mathcal{D}}_i(14)| = 4139$.

length and $m = 1$kg is the pendulum mass. The applied torque is restricted to $u \in [-2, 2]$ Nm. As implemented in OpenAI Gym, environment observations consist of the tuple $(\cos\theta, \sin\theta, \dot{\theta}) = (x, y, \dot{\theta})$. The angular velocity $\dot{\theta}$ is restricted to $[-8, 8]$. Starting from an arbitrary initial state, the goal is to balance the pendulum in the inverted position, i.e., $(\theta, \dot{\theta}) = (0, 0)$. The instantaneous cost function is

$$r(\theta[k], \dot{\theta}[k], u[k]) = \theta[k]^2 + 0.1\dot{\theta}[k]^2 + 0.001u[k]^2, \quad (37)$$

where $\theta[k]$ is wrapped between $[-\pi, \pi]$. Episodes terminate after 200 time steps.

As there does not exist a closed form expression for the optimal value function, we rely on state and action space discretization and a standard dynamic programming approach to compute a baseline for comparison with the agent value functions and performance. In particular, we uniformly quantize the state space to 1000 values for each state dimension and the action space to 50 actions to arrive at the reference value function shown in Figure 3b normalized to range $[0, 1]$.

For decentralized learning, we used the network in Figure 1b. We used a single Gaussian kernel with $\sigma = 0.17$, and set $\gamma = 0.5$ for Algorithm 1 in our experiment. 6000 state/action pairs were sampled uniformly from the joint state and action space, and divided equally amongst the agents, giving 600 points to each, initially. The value functions for each agent were estimated via the kernel embedding method outlined in the Section 3. Figure 4 collects the normalized value functions for each agent with initial local datasets, and later at time $t = 14$. Due to computational complexity, the agent figures reflect value function estimates over a quantized grid of only 75 values per state dimension, and 25 uniformly spaced action values. As the plots show, consensus reduces the variation in value function estimation across agents, and refines each estimate close to the reference in Figure 3b.

Agent	1	2	3	4	5	DP Ref
Mean	-284	-233	-249	-263	-251	**-176**
Median	-250	-241	-245	-251	-242	-134
Agent	6	7	8	9	10	DP Ref
Mean	-226	-240	-212	-253	-231	**-176**
Median	-239	-245	**-129**	-234	-131	-134

Table 1: 100 episode (rounded) aggregate performance statistics for reference and individual agent controllers and reference dynamic programming based controller at time $t = 14$.

There is no particular performance based threshold for us to declare that the inverted pendulum environment is solved. Therefore, we compare the performance of our agents to the reference in terms of performance statistics over 100 episodes. Due to the arbitrary initial state, per episode scores can vary widely, even in the case where a given policy successfully brings the pendulum to the goal position. Roughly speaking, a score of approximately -400 or higher usually indicates that the pendulum was brought upright near the goal position for a significant portion of the episode. As Table 1 shows, each agent effectively learns the task, and agent 8 actually exceeds the median performance of the reference.

8 Concluding Remarks

In this paper, we have proposed a coherence-based compressed decentralized learning algorithm for CME operators. We showed that our novel consensus algorithm preserves the average of the approximations of the covariance operators across the agents in a network. In addition, the corresponding dictionaries maintained by agents and iterative dynamics of approximations of the CME operators remain stable over time. We applied our consensus algorithm to distributed approximation of eigenfunctions of kernel transfer operators in a pair of oscillator systems and approximate value iteration in a benchmark reinforcement learning task. Our future directions include asymptotic analysis of agent and aggregate operator estimates. While we have proven the boundedness of agent estimates, it remains to show whether this long-term behavior can be characterized in term of, e.g., fixed points or periodic orbits. We will likewise investigate the asymptotic error of individual and aggregate agent estimates from the true kernel operators. Finally, we plan to extend our approach to the online streaming setting, where, beyond their initial data sets agents collect and share additional data as the consensus procedure progresses.

Acknowledgments

This work was supported by the NSF-EPCN-2031570 grant, NSF-CPS-2038775 grant and C3.ai Digital Transformation Institute. The authors thank the anonymous reviewers for their insightful comments. We also thank Prof. Tamer Başar at UIUC for his inputs on quantized consensus that informed both our algorithm design and the mathematical analysis.

References

Aysal, T. C.; and Barner, K. E. 2010. Convergence of consensus models with stochastic disturbances. *IEEE Transactions on Information Theory*, 56(8): 4101–4113.

Berlinet, A.; and Thomas-Agnan, C. 2011. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media.

Bobade, P.; Panagou, D.; and Kurdila, A. J. 2019. Multi-agent adaptive estimation with consensus in reproducing kernel Hilbert spaces. In *2019 18th European Control Conference (ECC)*, 572–577. IEEE.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym. *arXiv preprint arXiv:1606.01540*.

Carli, R.; Fagnani, F.; Frasca, P.; and Zampieri, S. 2010. Gossip consensus algorithms via quantized communication. *Automatica*, 46(1): 70–80.

El Chamie, M.; Liu, J.; and Başar, T. 2016. Design and analysis of distributed averaging with quantized communication. *IEEE Transactions on Automatic Control*, 61(12): 3870–3884.

Engel, Y.; Mannor, S.; and Meir, R. 2002. Sparse online greedy support vector regression. In *European Conference on Machine Learning*, 84–96. Springer.

Forero, P. A.; Cano, A.; and Giannakis, G. B. 2010. Consensus-Based Distributed Support Vector Machines. *Journal of Machine Learning Research*, 11(5).

Frasca, P.; Carli, R.; Fagnani, F.; and Zampieri, S. 2009. Average consensus on networks with quantized communication. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 19(16): 1787–1816.

Grünewälder, S.; Lever, G.; Baldassarre, L.; Patterson, S.; Gretton, A.; and Pontil, M. 2012. Conditional mean embeddings as regressors-supplementary. *arXiv preprint arXiv:1205.4656*.

Grunewalder, S.; Lever, G.; Baldassarre, L.; Pontil, M.; and Gretton, A. 2012. Modelling transition dynamics in MDPs with RKHS embeddings. *arXiv preprint arXiv:1206.4655*.

Hou, B.; Bose, S.; and Vaidya, U. 2021. Sparse Learning of Kernel Transfer Operators. In *2021 55th Asilomar Conference on Signals, Systems, and Computers*, 130–134. IEEE.

Kashyap, A.; Başar, T.; and Srikant, R. 2007. Quantized consensus. *Automatica*, 43(7): 1192–1203.

Kivinen, J.; Smola, A. J.; and Williamson, R. C. 2004. Online learning with kernels. *IEEE Transactions on Signal Processing*, 52(8): 2165–2176.

Klus, S.; Schuster, I.; and Muandet, K. 2020. Eigendecompositions of transfer operators in reproducing kernel Hilbert spaces. *Journal of Nonlinear Science*, 30(1): 283–315.

Koppel, A.; Paternain, S.; Richard, C.; and Ribeiro, A. 2018. Decentralized online learning with kernels. *IEEE Transactions on Signal Processing*, 66(12): 3240–3255.

Koppel, A.; Warnell, G.; Stump, E.; and Ribeiro, A. 2017. Parsimonious online learning with kernels via sparse projections in function space. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4671–4675. IEEE.

Lever, G.; Shawe-Taylor, J.; Stafford, R.; and Szepesvári, C. 2016. Compressed conditional mean embeddings for model-based reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Manton, J. H.; and Amblard, P.-O. 2015. A primer on reproducing kernel Hilbert spaces. *Foundations and Trends® in Signal Processing*, 8(1–2): 1–126.

Mezić, I. 2005. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1): 309–325.

Muandet, K.; Fukumizu, K.; Sriperumbudur, B.; and Schölkopf, B. 2016. Kernel mean embedding of distributions: A review and beyond. *arXiv preprint arXiv:1605.09522*.

Nedic, A.; Olshevsky, A.; Ozdaglar, A.; and Tsitsiklis, J. N. 2009. On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control*, 54(11): 2506–2517.

Richard, C.; Bermudez, J. C. M.; and Honeine, P. 2008. Online prediction of time series data with kernels. *IEEE Transactions on Signal Processing*, 57(3): 1058–1067.

Song, L.; Huang, J.; Smola, A.; and Fukumizu, K. 2009. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 961–968.

Szepesvári, C. 2010. Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, 4(1): 1–103.

Tolstikhin, I.; Sriperumbudur, B. K.; and Muandet, K. 2017. Minimax estimation of kernel mean embeddings. *The Journal of Machine Learning Research*, 18(1): 3002–3048.

Xiao, L.; Boyd, S.; and Lall, S. 2005. A scheme for robust distributed sensor fusion based on average consensus. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, 63–70. IEEE.