Neural Networks 117 (2019) 94–103

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Dynamic mode decomposition in vector-valued reproducing kernel Hilbert spaces for extracting dynamical structure among observables

Keisuke Fujii [a,b,*], Yoshinobu Kawahara [c,b]

[a] Graduate School of Informatics, Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Aichi, Japan
[b] Center for Advanced Intelligence Project, RIKEN, Furuedai, 6-2-3, Suita, Osaka, Japan
[c] Institute of Mathematics for Industry, Kyushu University, 744 Motooka, Nishi-ku, Fukuoka, Japan

HIGHLIGHTS

- Extraction methods of dynamical structures among observables are proposed.
- We incorporate structures among observables into dynamic mode decomposition.
- In formulation, we develop Koopman spectral analysis in vector-valued RKHSs.
- We investigate the empirical performance using synthetic and real-world datasets.
- We extracted dynamical information and spatial coherent patterns among observables.

ARTICLE INFO

Article history:
Received 30 August 2018
Received in revised form 21 March 2019
Accepted 25 April 2019
Available online 15 May 2019

Keywords:
Dynamical systems
Dimensionality reduction
Spectral analysis
Unsupervised learning

ABSTRACT

Understanding nonlinear dynamical systems (NLDSs) is challenging in a variety of engineering and scientific fields. Dynamic mode decomposition (DMD), which is a numerical algorithm for the spectral analysis of Koopman operators, has been attracting attention as a way of obtaining global modal descriptions of NLDSs without requiring explicit prior knowledge. However, since existing DMD algorithms are in principle formulated based on the concatenation of scalar observables, it is not directly applicable to data with dependent structures among observables, which take, for example, the form of a sequence of graphs. In this paper, we formulate Koopman spectral analysis for NLDSs with structures among observables and propose an estimation algorithm for this problem. This method can extract and visualize the underlying low-dimensional global dynamics of NLDSs with structures among observables from data, which can be useful in understanding the underlying dynamics of such NLDSs. To this end, we first formulate the problem of estimating spectra of the Koopman operator defined in vector-valued reproducing kernel Hilbert spaces, and then develop an estimation procedure for this problem by reformulating tensor-based DMD. As a special case of our method, we propose the method named as Graph DMD, which is a numerical algorithm for Koopman spectral analysis of graph dynamical systems, using a sequence of adjacency matrices. We investigate the empirical performance of our method by using synthetic and real-world data.

1. Introduction

Understanding nonlinear dynamical systems (NLDSs) or complex phenomena is a fundamental problem in various scientific and industrial fields. Complex systems are broadly defined as systems that comprise non-linearly interacting components (Boccaletti, Latora, Moreno, Chavez, & Hwang, 2006), in fields such as sociology, epidemiology, neuroscience, and physics (e.g., Bullmore & Sporns, 2009; Centola & Macy, 2007). As a method of obtaining a global modal description of NLDSs, operator- theoretic approaches have attracted attention such as in applied mathematics, physics and machine learning. One of the approaches is based on the composition operator (usually referred to as the Koopman operator (Koopman, 1931; Mezić, 2005)), which defines the time evolution of observation functions in a function space. A strength of this approach is that the spectral analysis of the operator can decompose the global property of NLDSs, because the analysis of NLDSs can be lifted to a linear but infinite dimensional regime. This approach can directly obtain dynamical structures such as frequency with delay/growth rate and the spatial coherences corresponding to the temporal information. Among several estimation methods, one of the most popular

[*] Corresponding author at: Graduate School of Informatics, Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Aichi, Japan.
E-mail address: keisuke.fujii.zh@riken.jp (K. Fujii).

https://doi.org/10.1016/j.neunet.2019.04.020
0893-6080/© 2019 Elsevier Ltd. All rights reserved.

algorithms for spectral analysis of the Koopman operator is dynamic mode decomposition (DMD) (Rowley, Mezić, Bagheri, Schlatter, & Henningson, 2009; Schmid, 2010), of which advantage is to extract such a modal description of NLDSs from data, unlike other unsupervised dimensionality reduction methods such as principal component analysis (PCA) for static data. DMD has been successfully applied in many real-world problems, such as image processing, neuroscience, and system control (e.g., Brunton, Johnson, Ojemann, & Kutz, 2016; Kutz, Fu, & Brunton, 2016). In a machine learning community, several algorithmic improvements have been accomplished by such as a formulation with reproducing kernels and in a Bayesian framework (e.g., Kawahara, 2016; Takeishi, Kawahara, Tabei and Yairi, 2017; Takeishi, Kawahara and Yairi, 2017). However, since conventional Koopman spectral analysis and DMDs are in principle formulated based on the concatenation of scalar observables, it is not directly applicable to data with dependent structures among observables, which take, for example, the form of a sequence of graphs.

The motivation of this paper is to understand NLDSs with dynamical structures among observables by extracting the low-dimensional global dynamics among observables. To this end, we develop a formulation of Koopman spectral analysis of NLDSs with structures among observables and propose an estimation algorithm for this problem. We first suppose that a sequence of matrices representing the dependency among observables (such as adjacency matrices of graphs) are observed as realizations of structures representing the relation of vector-valued observation function. Then, we formulate the problem of estimating the spectra of Koopman operators defined in reproducing kernel Hilbert spaces (RKHSs) endowed with kernels for vector-valued functions, called vector-valued RKHSs (vvRKHSs). Recently, there has been an increasing interest in kernels for vvRKHSs dealing with such as classification or regression problem with multiple outputs (e.g., Álvarez, Rosasco, Lawrence, et al., 2012; Micchelli & Pontil, 2005b and for the details, see Section 6). Thus, advantage or contribution of our method is that it can extract and visualize the dynamical structures among observables by incorporating the structure among variables in the vector-valued observation function into the DMD algorithm, which can be useful in understanding the fundamental dynamics behind spatiotemporal data with dependent structures. Second, we develop an estimation procedure from data by reformulating Tensor-based DMD (TDMD), which can compute DMD from tensor time-series data (Klus, Gelß, Peitz, & Schütte, 2018) without breaking tensor data structure (e.g., a sequence of adjacency matrices). We propose a more directly and stably computable TDMD than the previous algorithm.

Furthermore, as a special case of our method, we propose the method named as *Graph DMD*, which is a numerical algorithm for Koopman spectral analysis of graph dynamical systems (GDSs). GDSs are defined as spatially distributed units that are dynamically coupled according to the structure of a graph (Cliff, Prokopenko, & Fitch, 2016; Mortveit & Reidys, 2001). In mathematics, GDSs have been broadly studied such as in cellular automata (Mortveit & Reidys, 2007) and coupled NLDSs (Wu, 2005). Meanwhile, for graph sequence data, researchers have basically computed the graph (spatial) properties in each temporal snapshot (e.g., Bullmore & Sporns, 2009; Centola & Macy, 2007) or in a sliding window (e.g., Idé & Kashima, 2004) of the sequence data (for the details, see Section 6). However, these approaches would be difficult to extract the dynamical information directly from graph sequence data. We consider that our approach will solve this problem to understand the underlying global dynamics of GDSs.

Finally, we investigate the performance of our method with application to several synthetic and real-world datasets, including multi-agent simulation and sharing-bike data. These have the structures among observables, which represent, for example, the relation between agents (such as distance) and the traffic volume between locations, respectively.

The remainder of this paper is organized as follows. First, in Section 2, we briefly review the background of Koopman spectral analysis and DMD. Next, we describe the formulation in vvRKHS in Section 3, and reformulate DMD in a tensor form to estimate from data in Section 4. In Section 5, we propose Graph DMD for the analysis of GDSs. In Section 6, we describe related work. Finally, we show some experimental results using synthetic and real-world data in Section 7, and conclude this paper in Section 8.

2. Koopman spectral analysis and DMD

Here, we first briefly review Koopman spectral analysis, which is the underlying theory for DMD, and then describe the basic DMD procedure. First, we consider a NLDS: $\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t)$, where $\boldsymbol{x}_t$ is the state vector in the state space $\mathcal{M} \subset \mathbb{R}^p$ with time index $t \in \mathbb{T} := \mathbb{N}_0$ and $\boldsymbol{f}\colon \mathcal{M} \to \mathcal{M}$ is a (typically, nonlinear) state-transition function. *The Koopman operator*, which we denote by $\mathcal{K}$, is a linear operator acting on a scalar observation function $g\colon \mathcal{M} \to \mathbb{C}$ defined by

$$\mathcal{K}g = g \circ \boldsymbol{f}, \tag{1}$$

where $g \circ \boldsymbol{f}$ denotes the composition of g with $\boldsymbol{f}$ (Koopman, 1931). That is, it maps g to the new function $g \circ \boldsymbol{f}$. We assume that $\mathcal{K}$ has only discrete spectra. Then, it generally performs an eigenvalue decomposition: $\mathcal{K}\varphi_j(\boldsymbol{x}) = \lambda_j \varphi_j(\boldsymbol{x})$, where $\lambda_j \in \mathbb{C}$ is the jth eigenvalue (called *the Koopman eigenvalue*) and φ_j is the corresponding eigenfunction (called *the Koopman eigenfunction*). We denote the concatenation of scalar functions as $\boldsymbol{g} := [g_1, \ldots, g_m]^{\mathrm{T}}$. If each g_j lies within the space spanned by the eigenfunction φ_j, we can expand the vector-valued $\boldsymbol{g}\colon \mathcal{M} \to \mathbb{C}^m$ in terms of these eigenfunctions as $\boldsymbol{g}(\boldsymbol{x}) = \sum_{j=1}^{\infty} \varphi_j(\boldsymbol{x})\boldsymbol{\psi}_j$, where $\boldsymbol{\psi}_j$ is a set of vector coefficients called *the Koopman modes*. Through the iterative applications of $\mathcal{K}$, the following equation is obtained:

$$\boldsymbol{g}(\boldsymbol{x}_t) = (\boldsymbol{g} \circ \underbrace{\boldsymbol{f} \circ \cdots \circ \boldsymbol{f}}_{t})(\boldsymbol{x}_0) = \sum_{j=1}^{\infty} \lambda_j^t \varphi_j(\boldsymbol{x}_0)\, \boldsymbol{\psi}_j. \tag{2}$$

Therefore, λ_j characterizes the time evolution of the corresponding Koopman mode $\boldsymbol{\psi}_j$, i.e., the phase of λ_j determines its frequency and the magnitude determines the growth rate of its dynamics.

Among several possible methods to compute the above modal decomposition from data, DMD (Rowley et al., 2009; Schmid, 2010) is the most popular algorithm, which estimates an approximation of the decomposition in Eq. (2). Consider a finite-length observation sequence $\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_\tau$ $(\in \mathbb{C}^n)$, where $\boldsymbol{y} := \boldsymbol{g}(\boldsymbol{x}_t)$. Let $\boldsymbol{X} = [\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{\tau-1}]$ and $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_\tau]$. Then, DMD basically approximates it by calculating the eigendecomposition of matrix $\boldsymbol{F} = \boldsymbol{Y}\boldsymbol{X}^{\dagger}$, where $\boldsymbol{X}^{\dagger}$ is the pseudo-inverse of $\boldsymbol{X}$. The matrix $\boldsymbol{F}$ may be intractable to analyze directly when the dimension is large. Therefore, in the popular implementation of DMD such as exact DMD (Tu, Rowley, Luchtenburg, Brunton, & Kutz, 2014), a rank-reduced representation $\hat{\boldsymbol{F}}$ based on singular-value decomposition (SVD) is applied. That is, $\boldsymbol{X} \approx \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$ and $\hat{\boldsymbol{F}} = \boldsymbol{U}^*\boldsymbol{F}\boldsymbol{U} = \boldsymbol{U}^*\boldsymbol{Y}\boldsymbol{V}\boldsymbol{\Sigma}^{(-1)}$, where * is the conjugate transpose. Thereafter, we perform eigendecomposition of $\hat{\boldsymbol{F}}$ to obtain the set of the eigenvalues λ_j and eigenvectors $\boldsymbol{w}_j$. Then, we estimate the Koopman modes in Eq. (2): $\boldsymbol{\psi}_j = \lambda_j^{(-1)}\boldsymbol{Y}\boldsymbol{V}\boldsymbol{\Sigma}^{(-1)}\boldsymbol{w}_j$, which is called *DMD modes*.

3. Koopman spectral analysis in vvRKHSs for extracting dynamical structure among observables

Since the existing DMD algorithms basically estimate the spectra of Koopman operators defined in spaces of scalar observables, dependencies among observables are not taken into consideration. Therefore, they are in principle not applicable to analyze NLDSs with structure among observables. In this section, we formulate our method by considering the Koopman spectral analysis of such NLDSs in vvRKHSs endowed with kernels for vector-valued functions.

First, let $\mathcal{H}_{\boldsymbol{K}}$ be the vvRKHS endowed with a symmetric positive semi-definite kernel matrix $\boldsymbol{K}\colon \mathcal{M} \times \mathcal{M} \to \mathbb{R}^{m\times m}$ (Álvarez et al., 2012). That is, $\mathcal{H}_{\boldsymbol{K}}$ is a Hilbert space of functions $\boldsymbol{f}'\colon \mathcal{M} \to \mathbb{R}^m$, such that for every $\boldsymbol{c} \in \mathbb{R}^m$ and $\boldsymbol{x} \in \mathcal{M}$, $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}')\boldsymbol{c}$ as a function of $\boldsymbol{x}'$ belongs to $\mathcal{H}_{\boldsymbol{K}}$ and, moreover, $\boldsymbol{K}$ has the reproducing property

$$\left\langle \boldsymbol{f}', \boldsymbol{K}(\cdot, \boldsymbol{x})\boldsymbol{c}\right\rangle_{\boldsymbol{K}} = \boldsymbol{f}'(\boldsymbol{x})^\top \boldsymbol{c}, \tag{3}$$

where $\langle \cdot, \cdot \rangle_{\boldsymbol{K}}$ is the inner product in $\mathcal{H}_{\boldsymbol{K}}$.

In our formulation, we model the relation with a vector-valued observation function $\boldsymbol{g}$ for Koopman spectral analysis of NLDSs with structure among observables. Then, we assume that the components of $\boldsymbol{g}$ follow a Gaussian process given by a covariance kernel matrix. That is, the vector-valued observation function $\boldsymbol{g}\colon \mathcal{M} \to \mathbb{R}^m$ follows the Gaussian distribution

$$\boldsymbol{g}(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{x}), \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x})), \tag{4}$$

where $\boldsymbol{\mu} \in \mathbb{R}^m$ is a vector whose components are the mean functions $\mu_i(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathcal{M} \subset \mathbb{R}^p$ and $i = 1, \ldots, m$, and $\boldsymbol{K}$ is the above matrix-valued function. The entries $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x})_{i,j}$ in the matrix $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x})$ correspond to the covariances between the observables $g_i(\boldsymbol{x})$ and $g_j(\boldsymbol{x})$ for $i, j = 1, \ldots, m$. In our following formulation in the vvRKHS determined by $\boldsymbol{K}$, again, it is necessary for $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x})$ to be a symmetric positive semidefinite matrix. Practically, for example, we can use positive semidefinite (scalar-valued) kernels between observables as the components of $\boldsymbol{K}$.

Based on the above setting, we consider Koopman spectral analysis for NLDSs with structures among observables by extending the formulation of a scalar observation function of DMD with reproducing kernels (Kawahara, 2016) to that of relations within the vector-valued observable function in vvRKHSs (Álvarez et al., 2012). To this end, we first assume that the vector-valued observation function $\boldsymbol{g}$ is in the vvRKHS defined by $\boldsymbol{K}$, i.e., $\boldsymbol{g} \in \mathcal{H}_{\boldsymbol{K}}$. Then, the Koopman operator $\mathcal{K}_{\boldsymbol{K}}\colon \mathcal{H}_{\boldsymbol{K}} \to \mathcal{H}_{\boldsymbol{K}}$ defined by $\mathcal{K}_{\boldsymbol{K}}\boldsymbol{g} = \boldsymbol{g} \circ \boldsymbol{f}$, like Eq. (1), is a linear operator in $\mathcal{H}_{\boldsymbol{K}}$. Additionally, we denote by $\boldsymbol{\phi}_{\boldsymbol{c}}\colon \mathcal{M} \to \mathcal{H}_{\boldsymbol{K}}$ the feature map, i.e., $\boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{x}) = \boldsymbol{K}(\cdot, \boldsymbol{x})\boldsymbol{c}$ for any $\boldsymbol{c} \in \mathbb{R}^m$. According to Schrödl (2009), this is the second type of the feature map in the vvRKHS which directly maps to a Hilbert space $\mathcal{H}_{\boldsymbol{K}}$ and has been used in Evgeniou and Pontil (2004), Micchelli and Pontil (2005a) and Evgeniou, Micchelli, and Pontil (2005). Now, for every $\boldsymbol{c}$, we have the following proposition:

Proposition 3.1. *Assume $\boldsymbol{g} \in \mathcal{H}_{\boldsymbol{K}}$. Then, $(\mathcal{K}_{\boldsymbol{K}}\boldsymbol{g})(\boldsymbol{x}) = (\boldsymbol{g} \circ \boldsymbol{f})(\boldsymbol{x})$, which is in the case of the vector-valued observable in Eq. (1), equals to the following:*

$$(\mathcal{K}_{\boldsymbol{K}}{}^*\boldsymbol{\phi}_{\boldsymbol{c}})(\boldsymbol{x}) = (\boldsymbol{\phi}_{\boldsymbol{c}} \circ \boldsymbol{f})(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \mathcal{M}, \ \forall \boldsymbol{c} \in \mathbb{R}^m. \tag{5}$$

Proof. Since $\mathcal{K}_{\boldsymbol{K}}\boldsymbol{g} \in \mathcal{H}_{\boldsymbol{K}}$, we have, for any $\boldsymbol{c} \in \mathbb{R}^m$, $\mathcal{K}_{\boldsymbol{K}}\boldsymbol{g}(\boldsymbol{x})^\top \boldsymbol{c} = \left\langle \mathcal{K}_{\boldsymbol{K}}\boldsymbol{g}, \boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{x})\right\rangle_{\boldsymbol{K}} = \left\langle \boldsymbol{g}, \mathcal{K}_{\boldsymbol{K}}^*\boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{x})\right\rangle_{\boldsymbol{K}}$ for all $\boldsymbol{x} \in \mathcal{M}$. Similarly, $\boldsymbol{g} \circ \boldsymbol{f}(\boldsymbol{x})^\top \boldsymbol{c} = \left\langle \boldsymbol{g}, \boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{f}(\boldsymbol{x}))\right\rangle_{\boldsymbol{K}}$ because $\boldsymbol{f}(\boldsymbol{x}) \in \mathcal{M}$. As a result, since $(\mathcal{K}_{\boldsymbol{K}}\boldsymbol{g})(\boldsymbol{x}) = (\boldsymbol{g} \circ \boldsymbol{f})(\boldsymbol{x})$, we obtain $(\mathcal{K}_{\boldsymbol{K}}{}^*\boldsymbol{\phi}_{\boldsymbol{c}})(\boldsymbol{x}) = (\boldsymbol{\phi}_{\boldsymbol{c}} \circ \boldsymbol{f})(\boldsymbol{x})$. □

The adjoint of the Koopman operator $\mathcal{K}_{\boldsymbol{K}}^*$ (also known as the Perron–Frobenius operator) in this case acts as a linear operator in the space spanned by features $\boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathcal{M}$. Here, we denote the eigendecomposition of $\mathcal{K}_{\boldsymbol{K}}^*$ by $\mathcal{K}_{\boldsymbol{K}}^*\boldsymbol{\varphi}_j = \lambda_j\boldsymbol{\varphi}_j$.

For the practical implementation of the spectral decomposition of the linear operator, we usually need to project data onto directions that are effective in capturing the properties of data, like the standard DMD described in Section 2. In DMD with reproducing kernels (Kawahara, 2016), a kernel principal orthogonal direction is used for this purpose. However, such a projection is not straightforward for the current problem because the principal directions are not defined analogously for tensor data. Now, for a given finite time span $[0, \tau]$, we define $\mathcal{M}_1 := [\boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{x}_0), \ldots, \boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{x}_{\tau-1})]$ and $\mathcal{M}_2 := [\boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{x}_1), \ldots, \boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{x}_\tau)]$. Then, we adopt the projection onto some orthogonal directions $\boldsymbol{\nu}_j = \sum_{t=0}^{\tau-1}\alpha_{j,t}\boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{x}_t) = \mathcal{M}_1\boldsymbol{\alpha}_j$ for $j = 1, \ldots, p$, where the coefficients $\alpha_{j,t} \in \mathbb{R}$ and $\boldsymbol{\alpha}_j \in \mathbb{R}^\tau$ are computed based on a tensor decomposition (described in detail in Section 4). Let $\mathcal{U} = [\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_p]$ and $\mathcal{U} = \mathcal{M}_1\boldsymbol{\alpha}$ with the coefficient matrix $\boldsymbol{\alpha} \in \mathbb{R}^{\tau\times p}$. Since $\mathcal{M}_2 = \mathcal{K}_{\boldsymbol{K}}^*\mathcal{M}_1$, the projection of $\mathcal{K}_{\boldsymbol{K}}^*$ onto the space spanned by $\mathcal{U}$ is given as follows:

$$\hat{\boldsymbol{F}} = \mathcal{U}^*\mathcal{K}_{\boldsymbol{K}}^*\mathcal{U} = \boldsymbol{\alpha}^*(\mathcal{M}_1^*\mathcal{M}_2)\boldsymbol{\alpha}. \tag{6}$$

Then, if we let $\hat{\boldsymbol{F}} = \hat{\boldsymbol{T}}^{-1}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{T}}$ be the eigendecomposition of $\hat{\boldsymbol{F}}$, we obtain p DMD modes as $\boldsymbol{\psi}_j = \mathcal{U}\boldsymbol{b}_j$ for $j = 1, \ldots, p$, where $\boldsymbol{b}_j$ is the jth row of $\hat{\boldsymbol{T}}^{-1}$. The diagonal matrix $\hat{\boldsymbol{\Lambda}}$ comprising the eigenvalues represents the temporal evolution. To establish the above, we show the following theorem:

Theorem 3.2. *Assume that $\varphi_j(\boldsymbol{x})^\top \boldsymbol{c} = \left\langle \boldsymbol{\kappa}_j, \boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{x})\right\rangle_{\boldsymbol{K}}$ for some $\boldsymbol{\kappa}_j \in \mathcal{H}_{\boldsymbol{K}}$ and $\forall \boldsymbol{x} \in \mathcal{M}$. If $\boldsymbol{\kappa}_j$ is in the subspace spanned by $\boldsymbol{\nu}_j$, so that $\boldsymbol{\kappa}_j = \mathcal{U}\boldsymbol{a}_j$ for some $\boldsymbol{a}_j \in \mathbb{C}^p$ and $\mathcal{U} = [\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_p]$, then $\boldsymbol{a}_j$ is the left eigenvector of $\hat{\boldsymbol{F}}$ with eigenvalue λ_j, and also we have*

$$\boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{x}) = \sum_{j=1}^{p}(\varphi_j(\boldsymbol{x})^\top \boldsymbol{c})\boldsymbol{\psi}_j, \tag{7}$$

where $\boldsymbol{\psi}_j = \mathcal{U}\boldsymbol{b}_j$ and $\boldsymbol{b}_j$ is the right eigenvector of $\hat{\boldsymbol{F}}$.

Proof. Since $\mathcal{K}_{\boldsymbol{K}}^*\boldsymbol{\varphi}_j = \lambda_j\boldsymbol{\varphi}_j$, we have $\left\langle \boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{f}(\boldsymbol{x})), \boldsymbol{\kappa}_j\right\rangle_{\boldsymbol{K}} = \lambda_j \left\langle \boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{x}), \boldsymbol{\kappa}_j\right\rangle_{\boldsymbol{K}}$. Thus, from the assumption,

$$\left\langle \boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{f}(\boldsymbol{x})), \mathcal{U}\boldsymbol{a}_j\right\rangle_{\boldsymbol{K}} = \lambda_j \left\langle \boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{x}), \mathcal{U}\boldsymbol{a}_j\right\rangle_{\boldsymbol{K}}. \tag{8}$$

By evaluating at $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{\tau-1}$ and then stacking, we have $(\mathcal{U}\boldsymbol{a}_j)^*\mathcal{M}_2 = \lambda_j(\mathcal{U}\boldsymbol{a}_j)^*\mathcal{M}_1$. If we multiply $\boldsymbol{\alpha}$ from the right-hand side, this gives

$$\boldsymbol{a}_j^*\boldsymbol{\alpha}^*\mathcal{M}_1^*\mathcal{M}_2\boldsymbol{\alpha} = \lambda_j\boldsymbol{a}_j^*. \tag{9}$$

Since $\boldsymbol{\alpha}^*\mathcal{M}_1^*\mathcal{M}_2\boldsymbol{\alpha} = \hat{\boldsymbol{F}}$, this means $\boldsymbol{a}_j$ is the left eigenvector of $\hat{\boldsymbol{F}}$ with eigenvalue λ_j. Let $\boldsymbol{b}_j$ be the right eigenvector of $\hat{\boldsymbol{F}}$ with eigenvalue λ_j and the corresponding left eigenvector $\boldsymbol{a}_j$. Assuming these have been normalized so that $\boldsymbol{a}_i^*\boldsymbol{b}_j = \delta_{ij}$, then any vector $\boldsymbol{h} \in \mathbb{C}^p$ can be written as $\boldsymbol{h} = \sum_{j=1}^{p}(\boldsymbol{a}_j^*\boldsymbol{h})\boldsymbol{b}_j$. Applying this to $\mathcal{U}^*\boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{x})$ gives

$$\mathcal{U}^*\boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{x}) = \sum_{j=1}^{p}(\boldsymbol{a}_j^*\mathcal{U}^*\boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{x}))\boldsymbol{b}_j = \sum_{j=1}^{p}(\varphi_j(\boldsymbol{x})^\top \boldsymbol{c})\boldsymbol{b}_j. \tag{10}$$

Since $\boldsymbol{b}_j = \mathcal{U}^*\boldsymbol{\psi}_j$, this proves Eq. (7). □

The assumptions in the theorem mean that the data are sufficiently rich and thus a set of the orthogonal direction $\mathcal{U}$ gives a good approximation of the representation with the eigenfunctions of $\mathcal{K}_{\boldsymbol{K}}^*$. As in the case of Eq. (2), by the iterative applications of $\mathcal{K}_{\boldsymbol{K}}^*$, we obtain

$$\boldsymbol{\phi}_{\boldsymbol{c}}(\boldsymbol{x}_t) = \sum_{j=1}^{p}\lambda_j^t(\varphi_j(\boldsymbol{x}_0)^\top \boldsymbol{c})\boldsymbol{\psi}_j. \tag{11}$$

Thus, this theorem gives the connection between the above eigen-values / -vectors and the Koopman eigen-values / -functions.

In summary, the formulation first needs the sequence of the kernel matrices $\boldsymbol{K}(\boldsymbol{x}_t, \boldsymbol{x}_t)$ for $t = 0, \dots, \tau$ and then obtains the Koopman spectra of NLDSs with structures among observables by the decomposition of the feature map $\boldsymbol{\phi_c} = \boldsymbol{K}(\cdot, \boldsymbol{x})\boldsymbol{c}$, described in Eq. (11). From the above claims in the vvRKHS, it is seemingly necessary to give some $\boldsymbol{c}$ for $\boldsymbol{\phi_c}(\boldsymbol{x}_t)$ for its implementation. However, we do not require to give $\boldsymbol{c}$ because we do not need to directly compute $\boldsymbol{\phi_c} = \boldsymbol{K}(\cdot, \boldsymbol{x})\boldsymbol{c}$ but just need to compute the realization of $\mathcal{U}$ from the observed data. Concretely, for an implementation of the above analysis, we first regard the given or calculated matrices as a realization of the structure of the kernel matrices $\boldsymbol{K}(\boldsymbol{x}_t, \boldsymbol{x}_t)$ (see Section 4). We denote the realized matrices as $\boldsymbol{A}_t \in \mathbb{R}^{m\times m}$ for $t = 0, \dots, \tau$ or the tensor as $\mathcal{A} \in \mathbb{R}^{m\times m\times(\tau+1)}$. Second, we need to compute a projected matrix in the space spanned by the columns of $\mathcal{U}$ (see 4.2 and Appendix A for the relation), and then DMD solution $\hat{\boldsymbol{F}} \in \mathbb{R}^{p\times p}$ and DMD modes $\boldsymbol{\psi}_j \in \mathbb{C}^{m\times m}$ for $j = 1, \dots, p$ (see 4.3 and 4.4). In the next section, we develop the procedure by reformulating TDMD for computing these quantities from data.

4. Reformulated tensor-based DMD

Here, we reformulate TDMD by Klus et al. (2018) as an estimation algorithm for the above formulation using the sequence of kernel matrices $\boldsymbol{K}(\boldsymbol{x}_t, \boldsymbol{x}_t)$ for $t = 0, \dots, \tau$, i.e., to calculate the above quantities without breaking the dependent structure among observables. We first review the tensor-train (TT) format in Section 4.1, and then compute the projected matrix and the compute pseudo-inverse of a tensor in Section 4.2. Next, we reformulate TDMD (we call it *reformulated TDMD*) in Section 4.3 and finally describe DMD for our problem, i.e., NLDSs with structures among observables in Section 4.4. Note that although TDMD is applicable for analyzing higher-order complex dynamical systems, our problem considers a sequence of the matrices $\mathcal{A} \in \mathbb{R}^{m\times m\times(\tau+1)}$ as an input tensor, which is a sequence of the realization of the $\boldsymbol{K}(\boldsymbol{x}_t, \boldsymbol{x}_t)$'s structure.

4.1. TT-format

In general, it is known that analyzing high-dimensional data becomes infeasible due to the so-called curse of dimensionality. This could be moderated by exploiting low-rank tensor approximation approaches. Several tensor formats such as the canonical format, Tucker format, and TT-format have been developed for this purpose (see e.g., Grasedyck, Kressner, & Tobler, 2013). Among these formats, the TT-format is known to be relatively stable and scalable for high-order tensors compared with the other formats (Oseledets, 2011).

Here, we review the TT-format. Let $\mathcal{A} \in \mathbb{C}^{n_1\times\cdots\times n_d}$ be an order-d tensor, where n_l denotes the dimensionality of the lth mode for $l = 1, \dots, d$ (called *full-format*). In TT-decomposition (see Oseledets, 2011), $\mathcal{A}$ is decomposed into d core tensors $\mathcal{A}^{(l)} \in \mathbb{C}^{r_{l-1}\times n_l\times r_l}$, where $r_0 = r_d = 1$. r_l is called TT-rank, which controls the complexity of TT decomposition. For an elementary expression, any element of $\mathcal{A}$ is given by $\mathcal{A}_{i_1,\dots,i_d} = \sum_{k_0=1}^{r_0} \cdots \sum_{k_d=1}^{r_d} \mathcal{A}^{(1)}_{k_0,i_1,k_1} \cdot \dots \cdot \mathcal{A}^{(d)}_{k_{d-1},i_d,k_d}$, where the subscripts of the tensors denote the indices. Moreover, for two vectors $\boldsymbol{v} \in \mathbb{C}^{n_1}$ and $\boldsymbol{w} \in \mathbb{C}^{n_2}$, the tensor product $\boldsymbol{v} \otimes \boldsymbol{w} \in \mathbb{C}^{n_1\times n_2}$ is given by $(\boldsymbol{v} \otimes \boldsymbol{w})_{i,j} = (\boldsymbol{v} \cdot \boldsymbol{w}^\top)_{i,j} = v_i \cdot w_j$. Using the tensor product, the whole tensor can then be represented as $\mathcal{A} = \sum_{k_0=1}^{r_0} \cdots \sum_{k_d=1}^{r_d} \mathcal{A}^{(1)}_{k_0,:,k_1} \otimes \cdots \otimes \mathcal{A}^{(d)}_{k_{d-1},:,k_d}$, where colons are used to indicate all components of the mode, e.g., $\mathcal{A}^{(l)}_{k_{i-1},:,k_i} \in \mathbb{C}^{n_l}$.

To describe the matricizations and vectorizations (also called *tensor unfoldings*) for efficient computation, let $\mathcal{A}_{i_1,\dots,i_l,:,i_{l+1}\dots,i_d}$ denote an n_l-dimensional vector called the mode-l fiber, where $1 \le l \le d-1$. For the two ordered subsets $N' = \{n_1, \dots, n_l\}$ and $N'' = \{n_{l+1}, \dots, n_d\}$ of $N = \{n_1, \dots, n_d\}$, the matricization of $\mathcal{A}$ with respect to N' and N'' is denoted by $\mathcal{A}\Big|^{N''}_{N'} \in \mathbb{C}^{(n_1\cdot\ldots\cdot n_l)\times(n_{l+1}\cdot\ldots\cdot n_d)}$, which is defined by concatenating the mode-l fibers of $\mathcal{A}$. In the special case with $N' = N$ and $N'' = \varnothing$, the vectorization of $\mathcal{A}$ is given by $\mathrm{vec}(\mathcal{A}) \in \mathbb{C}^{n_1\cdot\ldots\cdot n_d}$.

4.2. Projected matrix and modified pseudo-inverse for TT-format

In this section, before reformulating the TDMD in TT-format, we modify the computation of the pseudo-inverse of a tensor in Klus et al. (2018) and then obtain the projected matrix in the space spanned by the columns of $\mathcal{U}$ in Section 3. Note that although our problem considers a sequence of matrices $\mathcal{A} \in \mathbb{R}^{m\times m\times(\tau+1)}$ as an input tensor, TDMD is applicable for analyzing higher-order complex dynamical systems. For TDMD, consider τ snapshots of d-dimensional tensor trains $\mathcal{X}, \mathcal{Y} \in \mathbb{C}^{n_1\times\cdots\times n_d\times\tau}$, where $\mathcal{X}_{:,\dots,:,i+1} \in \mathbb{C}^{n_1\times\cdots\times n_d}$ for $i = 0, \dots, \tau-1$ and $\mathcal{Y}_{:,\dots,:,i}$ for $i = 1, \dots, \tau$. Let $r_0, \dots, r_{d+1}$ and $s_0, \dots, s_{d+1}$ be the TT-ranks of $\mathcal{X}$ and $\mathcal{Y}$, respectively. Now, let $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{C}^{n_1\cdot\cdots\cdot n_d\times\tau}$ be the specific matricizations of $\mathcal{X}$ and $\mathcal{Y}$, where we contract the dimensions $n_1, \dots, n_d$ such that every column of $\boldsymbol{X}$ and $\boldsymbol{Y}$ is the vectorization of the corresponding $\tau = n_{d+1}$ snapshot, respectively.

To efficiently compute TDMD only with matrix products (without any tensor products), we first perform TT-decomposition of $\mathcal{X}$ and matricize to $\boldsymbol{X}$ as

$$\boldsymbol{X} = \boldsymbol{M}\boldsymbol{\Sigma}\boldsymbol{N}, \tag{12}$$

where $\boldsymbol{M} = \left(\sum_{k_0=1}^{r_0} \cdots \sum_{k_{d-1}=1}^{r_{d-1}} \mathcal{X}^{(1)}_{k_0,:,k_1} \otimes \cdots \otimes \mathcal{X}^{(d)}_{k_{d-1},:,:}\right)\Big|^{r_d}_{n_1,\dots,n_d}$, $\boldsymbol{N} = \left(\mathcal{X}^{(d+1)}_{:,:,k_{d+1}}\right)\Big|^{\tau}_{r_d}$, and $\boldsymbol{\Sigma}$ is a diagonal matrix with singular values in its diagonal elements computed by the SVD of $\mathcal{X}^{(d)}\Big|^{r_d}_{r_{d-1},n_d}$. $\boldsymbol{N}$ is equivalent to the last core $\mathcal{X}^{(d+1)}$, which is a matrix because $r_{d+1} = k_{d+1} = 1$. Note that this is similar to SVD in the matrix form, but SVD and this matricization after TT-decomposition are completely different. $\boldsymbol{M} \in \mathbb{C}^{n_1\cdots n_d\times r_d}$ computed by the first d core of $\mathcal{X}$ is left-orthogonal[1] due to the procedure of TT-decomposition algorithm (Oseledets, 2011), and reflects some part of tensor structure of $\mathcal{A}$ when folding in full-format. In our problem for NLDSs with dependent structures among observables, $\boldsymbol{M} = \boldsymbol{X}\boldsymbol{N}^\dagger\boldsymbol{\Sigma}^{-1} \in \mathbb{R}^{m^2\times r_d}$ for $\boldsymbol{X} \in \mathbb{R}^{m^2\times\tau}$ works as the projected matrix in the space spanned by the columns of $\mathcal{U} = \mathcal{M}_1\boldsymbol{\alpha}$ in Section 3. For the details of the relation, see Appendix A.

Next, we claim that the pseudo-inverse $\boldsymbol{X}^\dagger$ for the computations of the following TDMD is computed as shown in the following proposition[2]:

[1] In general, a matrix $\boldsymbol{A}$ is left-orthonormal if $\boldsymbol{A}^*\boldsymbol{A} = \boldsymbol{I}$ and right-orthonormal if $\boldsymbol{A}\boldsymbol{A}^* = \boldsymbol{I}$.

[2] The computation of the pseudo-inverse $\boldsymbol{X}^\dagger$ in Klus et al. (2018) is described with the left and right-orthonormalization of the cores of $\mathcal{X}$ including QR decompositions in a mathematically general way. However, when considering TDMD (i.e., $l = d-1$ and $\boldsymbol{N}$ being a matrix), the proposed algorithm with the pseudo-inverse of $\boldsymbol{N}$ using SVD (without QR decomposition) is more directly and stably computable than the previous algorithm. The difference in the computational efficiency between them depends on the problem such as the TT-ranks and dimensions of the tensor.

Proposition 4.1. *Assume that $\boldsymbol{X} \in \mathbb{C}^{n_1 \cdot \dots \cdot n_d \times \tau}$ matricized from $\mathcal{X} \in \mathbb{C}^{n_1 \times \dots \times n_d \times \tau}$ is decomposed as in Eq. (12). Then, the pseudo-inverse $\boldsymbol{X}^\dagger$ is given by*

$$\boldsymbol{X}^\dagger = \boldsymbol{N}^\dagger \, \boldsymbol{\Sigma}^{-1} \boldsymbol{M}^*. \tag{13}$$

Proof. Although $\boldsymbol{M}$ is left-orthogonal as mentioned above, the last core $\boldsymbol{N} = \mathcal{X}^{(d+1)} \in \mathbb{C}^{r_d \times n_{d+1}}$ is not right-orthogonal, i.e., $\boldsymbol{N} \cdot \boldsymbol{N}^* \neq \boldsymbol{I}$. Then, we can use the pseudo-inverse matrix $\boldsymbol{N}^\dagger \in \mathbb{C}^{n_{d+1} \times r_d}$, i.e., $\boldsymbol{X}^\dagger = \boldsymbol{N}^\dagger \, \boldsymbol{\Sigma}^{-1} \boldsymbol{M}^*$. Since $\boldsymbol{M}^* \cdot \boldsymbol{M} = \boldsymbol{I}$ and $\boldsymbol{N} \cdot \boldsymbol{N}^\dagger = \boldsymbol{I}$, it follows that the pseudo-inverse $\boldsymbol{X}^\dagger$ satisfies the necessary and sufficient conditions for the pseudo-inverse, i.e., it satisfies the following four equations:

$$\boldsymbol{X}\boldsymbol{X}^\dagger\boldsymbol{X} = \boldsymbol{M}\,\boldsymbol{\Sigma}\,\boldsymbol{N} \cdot \boldsymbol{N}^\dagger \, \boldsymbol{\Sigma}^{-1} \boldsymbol{M}^* \cdot \boldsymbol{M}\,\boldsymbol{\Sigma}\,\boldsymbol{N} = \boldsymbol{X}, \tag{14}$$

$$\boldsymbol{X}^\dagger\boldsymbol{X}\boldsymbol{X}^\dagger = \boldsymbol{N}^\dagger \, \boldsymbol{\Sigma}^{-1} \boldsymbol{M}^* \cdot \boldsymbol{M}\,\boldsymbol{\Sigma}\,\boldsymbol{N} \cdot \boldsymbol{N}^\dagger \, \boldsymbol{\Sigma}^{-1} \boldsymbol{M}^* = \boldsymbol{X}^\dagger, \tag{15}$$

$$(\boldsymbol{X}\boldsymbol{X}^\dagger)^* = (\boldsymbol{M}\,\boldsymbol{M}^*)^* = \boldsymbol{M}\,\boldsymbol{M}^* = \boldsymbol{X}\boldsymbol{X}^\dagger, \tag{16}$$

$$(\boldsymbol{X}^\dagger\boldsymbol{X})^* = (\boldsymbol{N}^\dagger\,\boldsymbol{N})^* = \boldsymbol{N}^\dagger\,\boldsymbol{N} = \boldsymbol{X}^\dagger\boldsymbol{X}. \tag{17}$$

For the fourth equation, we use the property of pseudo-inverse $(\boldsymbol{N}^\dagger\,\boldsymbol{N})^* = \boldsymbol{N}^\dagger\,\boldsymbol{N}$. □

4.3. Reformulated TDMD

4.3.1. TDMD solution

Using similar matricizations of $\mathcal{X}$, we can also represent the tensor unfolding $\boldsymbol{Y}$ as a matrix product, i.e.,

$$\boldsymbol{Y} = \left. \left(\sum_{l_0=1}^{s_0} \cdots \sum_{l_{d-1}=1}^{s_{d-1}} \mathcal{Y}^{(1)}_{l_0,:,l_1} \otimes \cdots \otimes \mathcal{Y}^{(d)}_{l_{d-1},:,:} \right) \right|_{n_1,\dots,n_d}^{s_{d+1}} \cdot \mathcal{Y}^{(d+1)} \Big|_{s_{d+1}}^{\tau} = \boldsymbol{P}\,\boldsymbol{Q}. \tag{18}$$

We abbreviate the indices of $\mathcal{Y}^{(d+1)}_{:,:,k_{d+1}}$ as $\mathcal{Y}^{(d+1)}$ because $r_{d+1} = k_{d+1} = 1$. Note that we do not require any special property of the tensor cores of $\mathcal{Y}$. Combining the representations of $\boldsymbol{X}^\dagger$ and $\boldsymbol{Y}$ and generalizing the basic DMD procedure in Section 2 to the tensor form, we can express the rank-reduced DMD solution $\hat{\boldsymbol{F}} \in \mathbb{C}^{r_{d+1} \times r_{d+1}}$ (equivalent to $\hat{\boldsymbol{F}}$ of Graph DMD in Section 3) as

$$\hat{\boldsymbol{F}} = \boldsymbol{M}^*\boldsymbol{Y} \cdot \boldsymbol{X}^\dagger\boldsymbol{M} = \boldsymbol{M}^* \cdot \boldsymbol{P}\,\boldsymbol{Q} \cdot \boldsymbol{N}^\dagger \, \boldsymbol{\Sigma}^{-1}. \tag{19}$$

To compute $\hat{\boldsymbol{F}}$ in Eq. (19) , we bypass this computational cost by splitting Eq. (19) into different parts. First, we consider that in the rank-reduced $\boldsymbol{M}^* \cdot \boldsymbol{P} \in \mathbb{R}^{r_d \times s_d}$, any entry is given by

$$(\boldsymbol{M}^* \cdot \boldsymbol{P})_{i,j} = \sum_{k_0=1}^{r_0} \cdots \sum_{k_{d-1}=1}^{r_{d-1}} \sum_{l_0=1}^{s_0} \cdots \sum_{l_{d-1}=1}^{s_{d-1}} \left(\mathcal{X}^{(1)}_{k_0,:,k_1} \right)^T \times \mathcal{Y}^{(1)}_{l_0,:,l_1} \cdot \ldots \cdot \left(\mathcal{X}^{(d)}_{k_{d-1},:,i} \right)^T \mathcal{Y}^{(d)}_{l_{d-1},:,j}. \tag{20}$$

This is based on the following computation: $\mathrm{vec}(\mathcal{X})^T \cdot \mathrm{vec}(\mathcal{Y}) = \Pi_{l=1}^{d} \left(\mathcal{X}^{(l)} \right)^T \cdot \mathcal{Y}^{(l)}$. In this way, we can compute $\boldsymbol{M}^* \cdot \boldsymbol{P}$ without leaving the TT-format, and we only have to reshape certain contractions of the TT-cores. This computation can be implemented efficiently using Algorithm 4 from Oseledets (2011). The result assumes that the TT-ranks of $\mathcal{X}$ and $\mathcal{Y}$ are small compared to the entire state space of these tensors. Indeed, the tensor ranks r_d and s_d are both bounded by the number of snapshots τ. Second, for $\boldsymbol{Q} \cdot \boldsymbol{N}^\dagger$ in Eq. (19) , we simply obtain

$$\boldsymbol{Q} \cdot \boldsymbol{N}^\dagger = \mathcal{Y}^{(d+1)} \Big|_{s_d}^{\tau} \cdot \left(\mathcal{X}^{(d+1)} \Big|_{r_d}^{\tau} \right)^\dagger. \tag{21}$$

In this computation, we do not need to convert any tensor products of the cores of $\mathcal{X}$ or $\mathcal{Y}$, into full tensors during our calculations.

4.3.2. TDMD mode

Next, we consider the computation of the DMD modes of $\hat{\boldsymbol{F}}$. If $\lambda_1, \dots, \lambda_p$ are the eigenvalues of $\hat{\boldsymbol{F}}$ corresponding to the eigenvectors $\boldsymbol{w}_1, \dots, \boldsymbol{w}_p \in \mathbb{C}^{r_{d+1}}$, then the vectorized DMD modes $\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_p \in \mathbb{C}^{n_1 \cdot \dots \cdot n_d}$ of $\boldsymbol{F}$ (as in Section 2) are given by $\boldsymbol{\varphi}_j = (1/\lambda_j) \cdot \boldsymbol{P}\,\boldsymbol{Q} \cdot \boldsymbol{N}^\dagger \, \boldsymbol{\Sigma}^{-1} \cdot \boldsymbol{w}_j$, for $j = 1, \dots, p$. Tensor representation $\mathcal{Z} \in \mathbb{C}^{n_1 \times \dots \times n_d \times p}$ including all DMD modes is given by

$$\mathcal{Z} = \sum_{l_0=1}^{s_0} \cdots \sum_{l_d=1}^{s_d} \mathcal{Y}^{(1)}_{l_0,:,l_1} \otimes \cdots \otimes \mathcal{Y}^{(d)}_{l_{d-1},:,l_d} \otimes \left(\boldsymbol{Q} \cdot \boldsymbol{N}^\dagger \, \boldsymbol{\Sigma}^{-1} \cdot \boldsymbol{W} \cdot \boldsymbol{\Lambda}^{-1} \right)_{l_d,:}, \tag{22}$$

again with $\mathrm{vec}(\mathcal{Z}_{:,\dots,:,j}) = \boldsymbol{\varphi}_j$ and $\boldsymbol{\Lambda}$ is a diagonal matrix arranging $\lambda_1, \dots, \lambda_p$.

The overall algorithm of the reformulated TDMD is shown in Algorithm 1. We can express the DMD modes using given tensor trains $\mathcal{X}$ and $\mathcal{Y}$, modifying just the last core. In this case, we benefit from not leaving the TT-representations of $\mathcal{X}$ and $\mathcal{Y}$. In other words, the bottleneck of this algorithm regarding scalability would be sequential SVDs in TT-decompositions of $\mathcal{X}$ and $\mathcal{Y}$.

Algorithm 1 Reformulated Tensor-based DMD

```
Input: $\mathcal{X}, \mathcal{Y} \in \mathbb{C}^{n_1 \cdot \dots \cdot n_d \times \tau}$
Output: dynamic mode tensor $\mathcal{Z}$ and eigenvalue matrix $\boldsymbol{\Lambda}$
$\boldsymbol{M}, \boldsymbol{\Sigma}, \boldsymbol{N} \leftarrow$ matricized after decomposition of $\mathcal{X}$;
$\boldsymbol{N}^\dagger \leftarrow$ pseudo-inverse of $\boldsymbol{N}$;
$\boldsymbol{P}, \boldsymbol{Q} \leftarrow$ matricized after decomposition of $\mathcal{Y}$;
$\hat{\boldsymbol{F}} \leftarrow (\boldsymbol{M}^* \cdot \boldsymbol{P})(\boldsymbol{Q} \cdot \boldsymbol{N}^\dagger)\boldsymbol{\Sigma}^{-1}$;
$\boldsymbol{\Lambda}, \boldsymbol{W} \leftarrow$ eigendecomposition of $\hat{\boldsymbol{F}}$;
$\mathcal{Z} \leftarrow \sum_{l_0=1}^{s_0} \cdots \sum_{l_d=1}^{s_d} \mathcal{Y}^{(1)}_{l_0,:,l_1} \otimes \cdots \otimes \mathcal{Y}^{(d)}_{l_{d-1},:,l_d} \otimes \left( \boldsymbol{Q} \cdot \boldsymbol{N}^\dagger \, \boldsymbol{\Sigma}^{-1} \cdot \boldsymbol{W} \cdot \boldsymbol{\Lambda}^{-1} \right)_{l_d,:}$;
return: $\mathcal{Z}$, $\boldsymbol{\Lambda}$;
```

4.4. DMD for NLDSs with structures among observables

In DMD for NLDSs with structures among observables, as a special case of the above reformulated TDMD in 4.3, we use a sequence of the matrices $\mathcal{A} \in \mathbb{R}^{m \times m \times (\tau+1)}$ as a sequence of the realization of the $\boldsymbol{K}(\boldsymbol{x}_t, \boldsymbol{x}_t)$'s structure for $t = 0, \dots, \tau$ in reformulated TDMD, i.e., $d = 2$ and $n_1 = n_2 = m$. Input tensors $\mathcal{X}$ and $\mathcal{Y}$ are created from $\mathcal{A}_{:,:,t}$ and $\mathcal{A}_{:,:,t+1}$ for $t = 0, \dots, \tau - 1$, respectively. As a result, we obtain DMD modes $\boldsymbol{\psi}_j \in \mathbb{C}^{m \times m}$ as in Section 3 by matricizing $\boldsymbol{\varphi}_j$ (or $\mathcal{Z}_{:,:,j}$) with eigenvalues λ_j. The overall algorithm is shown in Algorithm 2.

Algorithm 2 DMD for NLDSs with structures among observables

```
Input: sequence of the matrices $\mathcal{A} \in \mathbb{R}^{m \times m \times (\tau+1)}$
Output: dynamic mode matrix $\boldsymbol{\psi}_j \in \mathbb{R}^{m \times m}$ and eigenvalue $\lambda_j$
$\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{m \times m \times \tau} \leftarrow$ make tensors from $\mathcal{A}$;
$\boldsymbol{M}, \boldsymbol{\Sigma}, \boldsymbol{N} \leftarrow$ matricized after decomposition of $\mathcal{X}$;
$\boldsymbol{N}^\dagger \leftarrow$ pseudo-inverse of $\boldsymbol{N}$;
$\boldsymbol{P}, \boldsymbol{Q} \leftarrow$ matricized after decomposition of $\mathcal{Y}$;
$\hat{\boldsymbol{F}} \leftarrow (\boldsymbol{M}^* \cdot \boldsymbol{P})(\boldsymbol{Q} \cdot \boldsymbol{N}^\dagger)\boldsymbol{\Sigma}^{-1}$;
$\boldsymbol{\Lambda}, \boldsymbol{W} \leftarrow$ eigendecomposition of $\hat{\boldsymbol{F}}$;
$\mathcal{Z} \leftarrow \sum_{l_0=1}^{s_0} \sum_{l_1=1}^{s_1} \sum_{l_2=1}^{s_2} \mathcal{Y}^{(1)}_{l_0,:,l_1} \otimes \mathcal{Y}^{(2)}_{l_1,:,l_2} \otimes \left( \boldsymbol{Q} \cdot \boldsymbol{N}^\dagger \, \boldsymbol{\Sigma}^{-1} \cdot \boldsymbol{W} \cdot \boldsymbol{\Lambda}^{-1} \right)_{l_2,:}$;
return: $\boldsymbol{\psi}_j = \mathcal{Z}_{:,:,j}$, $\lambda_j = \boldsymbol{\Lambda}_{j,j}$;
```

5. Graph DMD

In this section, as a special case of DMD for NLDSs with structures among observables in 4.4, we propose the method named as *Graph DMD*, which is a numerical algorithm for Koopman spectral analysis of GDSs. According to the notation of Cliff et al. (2016), we consider an autonomous discrete-time weighted and undirected GDS defined as

$$G = (\mathcal{V}, \mathcal{E}, \boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{f}, \boldsymbol{g}, \boldsymbol{A}_t), \tag{23}$$

where $\mathcal{V} = \{V^1, \dots, V^m\}$ and $\mathcal{E} = \{E^1, \dots, E^l\}$ are the vertex and edge sets of a graph, respectively, fixed at each time $t \in \mathbb{T}$. $\boldsymbol{x}_t \in \mathcal{M} \subset \mathbb{R}^p$ for the GDS and $\boldsymbol{f}: \mathcal{M} \rightarrow \mathcal{M}$ is a (typically, nonlinear) state-transition function (i.e., $\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t)$). $\boldsymbol{y}_t \in \mathbb{R}^m$ are observed values that correspond to vertices and are given by $\boldsymbol{y}_t := \boldsymbol{g}(\boldsymbol{x}_t)$, where $\boldsymbol{g}: \mathcal{M} \rightarrow \mathbb{R}^m$ is a vector-valued observation function. $\boldsymbol{A}_t \in \mathbb{R}^{m \times m}$ is an adjacency matrix, whose component $a_{i,j,t}$ represents the weight on the edge between V^i and V^j at each time t. For example, the weight represents some traffic volume between the locations in networks or public transportations. Another example of the weights for undirected GDSs is the relation between moving agents (such as distances) in multi-agent systems (Couzin, Krause, James, Ruxton, & Franks, 2002; Fujii et al., 2016).

In Graph DMD, we consider a sequence of adjacency matrices $\boldsymbol{A}_t \in \mathbb{R}^{m \times m}$ for $t = 0, \dots, \tau$ or $\mathcal{A} \in \mathbb{R}^{m \times m \times (\tau+1)}$ as input. Here, we assume that the adjacency matrix $\boldsymbol{A}_t$ observed at each time is a realization of the structure of the kernel matrix $\boldsymbol{K}(\boldsymbol{x}_t, \boldsymbol{x}_t)$ in Section 3. That is, the weight of $\boldsymbol{A}_t$ is assumed to represent the correlation between the observables. Again, in our formulation in the vvRKHS determined by $\boldsymbol{K}(\boldsymbol{x}_t, \boldsymbol{x}_t)$, it is necessary for $\boldsymbol{K}(\boldsymbol{x}_t, \boldsymbol{x}_t)$ to be a symmetric positive semidefinite matrix (i.e., we consider an undirected graph). For an implementation of Graph DMD, we use a sequence of adjacency matrices $\mathcal{A} \in \mathbb{R}^{m \times m \times (\tau+1)}$ in DMD for NLDSs with structures among observables. That is, we only replace the sequence of the matrices in Algorithm 2 with a sequence of adjacency matrices. As a result, similarly in Algorithm 2, we obtain DMD modes $\boldsymbol{\psi}_j \in \mathbb{C}^{m \times m}$ with eigenvalues λ_j.

6. Relation to previous works

6.1. Dynamic mode decomposition

Spectral analysis (or decomposition) for analyzing dynamical systems is a popular approach aimed at extracting low-dimensional dynamics from data. DMD, originally proposed in fluid physics (Rowley et al., 2009; Schmid, 2010), has recently attracted attention also in other areas of science and engineering, including analysis of power systems (Susuki & Mezić, 2014), epidemiology (Proctor & Eckhoff, 2015), neuroscience (Brunton et al., 2016), image processing (Kutz et al., 2016; Takeishi, Kawahara, & Yairi, 2017a), controlled systems (Proctor, Brunton, & Kutz, 2016), and human behaviors (Fujii, Inaba, & Kawahara, 2017; Fujii, Kawasaki, Inaba, & Kawahara, 2018; Fujii, Takeishi, Kibushi, Kouzaki, & Kawahara, 2019). Moreover, there are several algorithmic variants to overcome the problem of the original DMD such as the use of nonlinear basis functions (Williams, Kevrekidis, & Rowley, 2015), a formulation in a reproducing kernel Hilbert space (Kawahara, 2016), in a supervised learning framework via multitask learning (Fujii & Kawahara, 2019), in a Bayesian framework (Takeishi, Kawahara et al., 2017), and using a neural network (Takeishi, Kawahara, & Yairi, 2017b). For interconnected systems, e.g., Susuki and Mezić (2011) computed Koopman modes of coupled swing dynamics in power systems, and Heersink, Warren, and Hoffmann (2017) proposed DMD for (simulated) interconnected control systems, which extends DMD with control (Proctor et al., 2016). Note that these are basically formulated without considering the structures among observables unlike our formulation described in this paper.

6.2. Vector-valued RKHSs

RKHSs of vector-valued function (vvRKHS), endowed with a matrix-valued or operator-valued kernel (Caponnetto, Micchelli, Pontil, & Ying, 2008), have attracted an increasing interest as the methods to deal with such as classification or regression problem with multiple outputs (e.g., Álvarez et al., 2012; Micchelli & Pontil, 2005b). In real-world problems, this approach has applied to such as image processing (Quang, Kang, & Le, 2010) and medical treatment effects (Alaa & van der Schaar, 2017). Gaussian processes for vector-valued functions have also been formulated using the covariance kernel matrix (Álvarez et al., 2012). We first formulated the modal decomposition methods in the vvRKHS with the assumption that the vector-valued observable follows Gaussian process. Other researchers performed spatiotemporal pattern extraction by spectral analysis of vector-valued observables using operator-valued kernel (Giannakis, Ourmazd, Slawinska, & Zhao, 2017), but did not formulate in vvRKHSs and directly extract the dynamical information about the dependent structure among observables.

6.3. Other algorithms for graph data

For signal processing of a graph, researchers have basically examined the graph property in the graph (spatial) domain such as using graph Laplacians (e.g., Chung, 1997), graph Fourier transforms (e.g., Taubin, 1995), and graph convolutional networks (e.g., Defferrard, Bresson, & Vandergheynst, 2016). Graph Laplacians and other extensions have been also used for regularization by utilizing data structures (e.g., Liu, Zha, Wang, Lu, & Tao, 2016). For the graph sequence data in several scientific fields (see Section 1), various analyses have been examined such as using topological variables (Bullmore & Sporns, 2009) and objective variables in simulation (Centola & Macy, 2007; Kuhlman et al., 2011) in each snapshot of the graph time series, or performed graph abnormality detection (Idé & Kashima, 2004) with the temporal sliding windows of the time series. A few methods have been directly applied to the graph time series such as using graph convolutional networks (e.g., Seo, Defferrard, Vandergheynst, & Bresson, 2017). Meanwhile, our method has advantages to directly extract the underlying low-dimensional dynamics of GDSs.

7. Experimental results

We conducted experiments to investigate the empirical performance of our method (for clarity, we called it Graph DMD in this section) using synthetic data in Section 7.1. Then, we examined the applications to extract and visualize specific spatiotemporal dynamics in a real-world bike-sharing system data in Section 7.2 and in fish-schooling simulation data as an example of unknown global dynamics in Section 7.3. Note that, as mentioned in Section 6, most of the conventional methods for a graph have basically extracted the graph property in the graph (spatial) domain. Meanwhile, our method directly extracts the underlying low-dimensional dynamics among observables, which cannot be estimated by these methods. Therefore, we did not compare conventional methods for a graph but compare the conventional DMD algorithms with our method.

7.1. Synthetic data

We first validated the performance of our method to extract the dynamical information on synthetic data. We generated a sequence of noisy adjacency matrix series $\boldsymbol{A}_t \in \mathbb{R}^{D \times D}$ using the following equations:

$$\boldsymbol{A}_t = 0.99^t \boldsymbol{A}_{m_1} + 0.9^t \boldsymbol{A}_{m_2} + \boldsymbol{e}_t, \tag{24}$$

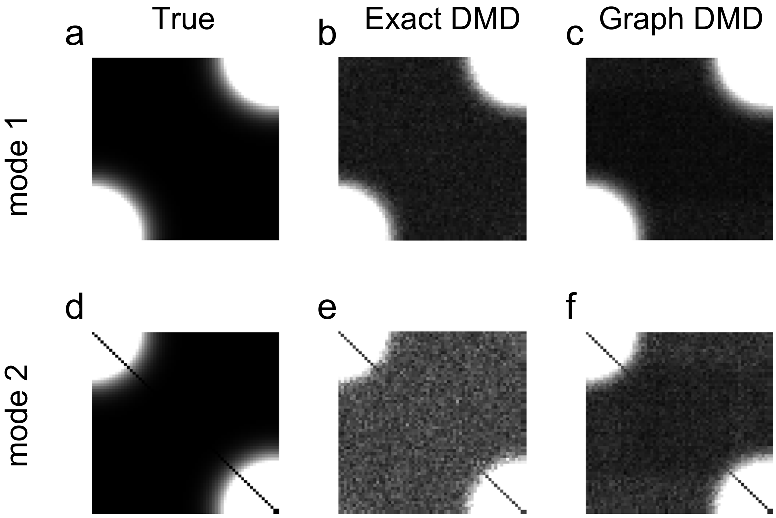

Fig. 1. Two spatial DMD modes for two temporal modes estimated by each method. (a) and (d): ground truth. (b) and (e): results of exact DMD. (c) and (f): results of Graph DMD.

where $\boldsymbol{A}_{m_1}, \boldsymbol{A}_{m_2} \in \mathbb{R}^{D\times D}$ (and $\boldsymbol{A}_t$ for every t) are the adjacency matrices shown in Fig. 1a and d, respectively. Black and white indicate lower and higher values, respectively. D was set to 64 and 256 for examining the effect of data dimension (Fig. 1 is for $D = 64$). Each element of $\boldsymbol{e}_t \in \mathbb{R}^{D\times D}$ is independently and identically sampled from a zero-mean Gaussian with variance 1e−02. In this case, the true spatial dynamic modes are $\boldsymbol{A}_{m_1}$ and $\boldsymbol{A}_{m_2}$, with the corresponding DMD eigenvalues 0.99 and 0.9 (mode 1 and mode 2), respectively. We here estimated the spatial and temporal modes from noisy data using our method (Graph DMD) and the exact DMD (Tu et al., 2014) as a baseline (described in Section 2). In Graph DMD (or reformulated Tensor-based DMD), TT-decomposition parameter ε (i.e., the tolerance in the successive SVD) is critical for estimating a few DMD mode such as in this case. The estimation performances are computed using the average values of 10 tasks.

The estimation results are shown in Fig. 1 and Table 1. The effect of the estimation errors for the two leading eigenvalue was evaluated by the relative errors defined by $\Delta|\lambda| = |\lambda - \tilde{\lambda}|/|\lambda|$, where λ is the estimated eigenvalue and $\tilde{\lambda}$ is the ground truth of the eigenvalues (0.99 and 0.9 for modes 1 and 2, respectively). Our proposed method with $\varepsilon = $ 1e−01 was more accurate than that with exact DMD. Note that in this experiment, the result of our method was the same as exact DMD when $\varepsilon \leq$ 1e−02 and our method extracted only one mode (i.e. one eigenvalue) when $\varepsilon >$ 1e−01. With respect to the size effect of the adjacency matrix, the larger the size, the higher is the estimation error because of the larger amount of noise. For the two leading spatial DMD modes, our method with $\varepsilon = $ 1e−02 decreased more noise (especially in Fig. 1f) than the exact DMD shown in Fig. 1e (the results of our method with $\varepsilon = $ 1e−02 are the same as those for the exact DMD). In addition, we confirmed that there were almost no differences in the eigenvalues between Graph DMD and the original TDMD (Klus et al., 2018) (< 1e−12 for all eigenvalues).

7.2. Bike-sharing data

One of the direct applications of our method is to extract the dynamical structure among observables. In some real-world datasets, we can use prior knowledge about the dynamics such as biological rhythms (e.g., a day, month, and year). Then, our method can extract the spatial (e.g., graph) coherent structure for the focusing dynamics (e.g., rhythm or frequency). Here, we extracted and illustrated graph (spatial) DMD modes with real-world bike-sharing system data. The bike-sharing data consisted of the numbers of bikes returned from one station to another in an hour in Washington D.C.[3] We collected a sum of the numbers of bikes transported between the two different stations for both directions for use as an undirected graph series. We selected 14 days from 2nd Sunday of every month of 2014 for 348 bike stations, and constructed the sequence of adjacency matrices $\mathcal{X} \in \mathbb{R}^{348\times 348\times 336}$. We consider that the relation between locations is stronger as the number of bikes increases. In this experiment, to obtain the smooth adjacency matrix series to extract dynamic properties, we summed up 14 days of data for every month and perform 12-point (i.e., half day) moving average. Fig. 2a shows an example of the preprocessed number of bikes between Lincoln memorial and three stations with a maximal number of bikes moving from/to Lincoln memorial. These seemed to be coherent and cycled at daily and weekly cycles.

Table 1
Estimation error of the two leading DMD eigenvalue for the numerical example. The entries show the relative errors $\Delta|\lambda|$ for different values of ε and D.

$\Delta\|\lambda\|$	Size 64 × 64		Size 256 × 256	
	Mode 1	Mode 2	Mode 1	Mode 2
DMD	1.49e−03	9.04e−03	1.85e−02	4.05e−01
$\varepsilon = $ 1e−02	1.49e−03	9.04e−03	1.85e−02	4.05e−01
$\varepsilon = $ 1e−01	9.33e−04	6.15e−03	1.85e−02	3.74e−01

Fig. 2b shows the eigenvalues estimated by Graph DMD. We confirm that most eigenvalues are on the unit circle, indicating that the dynamics were almost oscillators. Among these eigenvalues, we focused on the specific temporal modes of the traffic such as daily and weekly periodicity (Takeuchi, Kawahara, & Iwata, 2017), i.e., we selected $\omega = |\mathrm{Im}(\log(\lambda))|/\Delta t/(2\pi) = \{1/24, 1/168\}$, where λ is Graph DMD eigenvalue ($\Delta t = 1$ [hour]).

Fig. 2c and d shows the spatial (graph) pattern of the DMD mode for approximately $\omega = 1/24$ and $\omega = 1/168$ on the bike station map, respectively. Although the bike transportation near Lincoln memorial (left circle in Fig. 2c and d) shows a stronger spectrum for both daily and weekly periodicity, the bike transportation in a downtown area near Union Station (right circle in

[3] https://www.capitalbikeshare.com/.

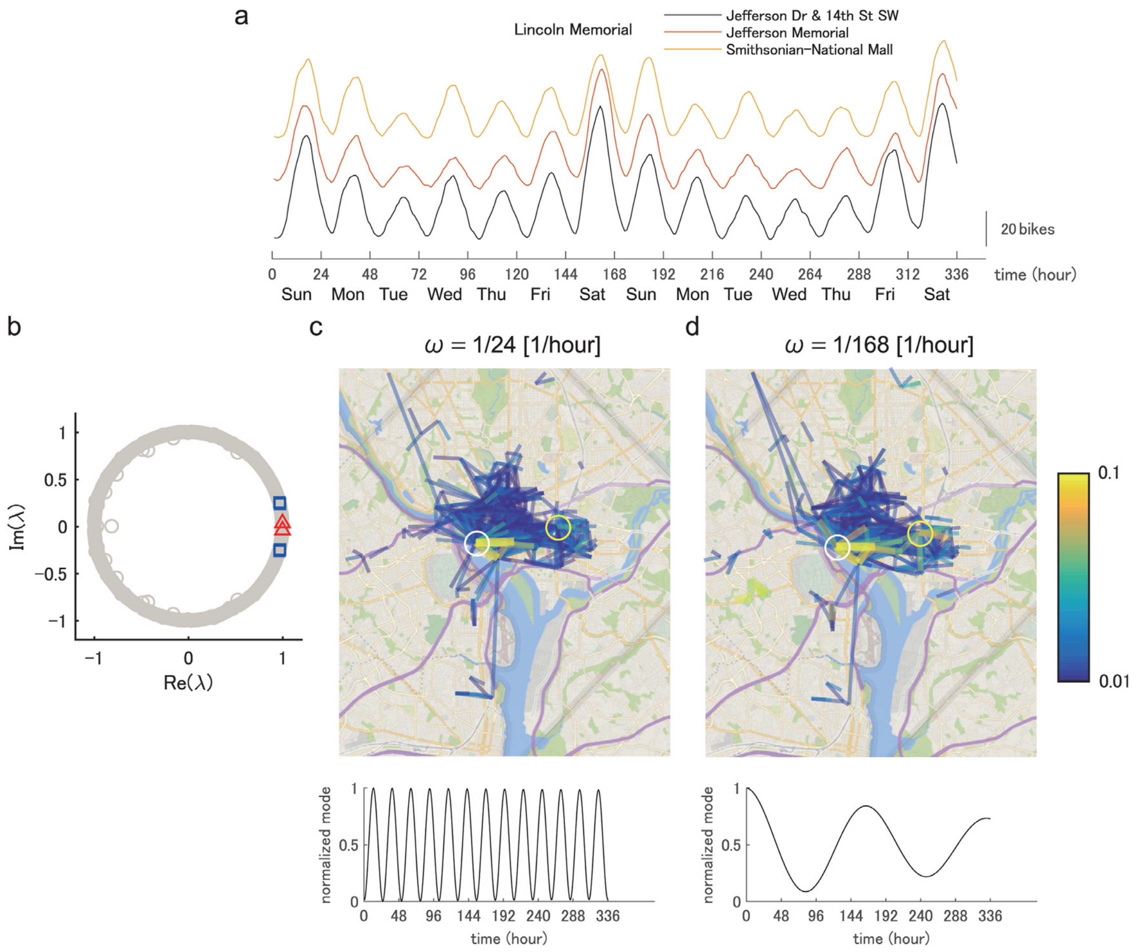

Fig. 2. Temporal and spatial modes of Graph DMD on the bike-sharing data of Washington D.C. (a) an example of the number of bikes between Lincoln memorial and three stations with a maximal number of bikes moving from/to Lincoln memorial. (b) a result of DMD eigenvalue. The blue square and red triangle indicate eigenvalues of approximately $\omega = 1/24$ and $\omega = 1/168$, respectively. (c) and (d): Graphical representation of the amplitude of spatial DMD modes in $\omega = 1/24$ and $\omega = 1/168$, respectively. The left and right circles indicate the Lincoln Memorial and Columbus Circle/Union Station, respectively. Lower time series in (c) and (d) are examples of the extracted temporal dynamics corresponding to the above spatial modes (visualized spatial modes are averaged among multiple modes). For improving the visibility, we used only an eigenvalue and set initial values 0 and 1 for (c) and (d), respectively.

Fig. 2c and d) for weekly periodicity shows a stronger spectrum than that for daily periodicity.

Overall, our method can extract the different spatial (graph) modes for specific temporal modes based on the dynamical structure. Note that in the formulation (again, in functional space), we also assume that the covariance matrix $\boldsymbol{K}$ is a symmetric positive semi-definite matrix-valued function. Numerically, however, as is the case of real-world data, we did not assume that the (symmetric) adjacency matrix $\boldsymbol{A}_t$ is positive semi-definite. In Appendix B, we proposed the alternative to modify them to positive semi-definite matrices (the results were similar to Fig. 2).

7.3. Fish-schooling model

Next, we evaluated our method using a example with unknown global dynamics, because in some real-world (especially biological) data, the true global spatiotemporal structure is sometimes unknown (Fujii et al., 2018; Hojo, Fujii, Inaba, Motoyasu, & Kawahara, 2018). For evaluation, here we used well-known collective motion models (Vicsek, Czirók, Ben-Jacob, Cohen, & Shochet, 1995) with simple local rules to generate multiple distinct group behavioral patterns (Fig. 3a): swarm, torus, and parallel behavioral shapes. The detailed configuration and simulation of the experiments are described in Appendix C and D, respectively. We used Gaussian kernels to create the sequences of adjacency matrices using inter-agent distance (for details, see Appendix E) because the local rules were applied based on the distance. First, the results in the temporal DMD mode, interpolating the discrete frequency spectra, exhibit a relatively wide spectrum for the swarm (Fig. 3b), a narrow spectrum for the torus (Fig. 3e) and parallel (Fig. 3g). Among these spectra, we focused on characteristic low- (0–2 Hz) and high-frequency (2–4 Hz) modes. The spectra in the swarm (Fig. 3c,d) and torus (Fig. 3f) show relatively stronger spectra nearer individuals, compared with that in the parallel (Fig. 3h). Thus, our method can visualize the observed interaction behaviors.

Although a direct and important application of Graph DMD is the extraction of the dynamical information for GDS, it can also perform embedding and recognition of GDSs using extracted features based on the dynamical structure. For embedding the distance matrix with DMD modes such as using multidimensional scaling (MDS), the components of the distance matrix depend on the problem. In this experiment, we compute the distance matrix between the temporal frequency modes by the alignment of the number of dimensions from larger frequencies, because of the results shown in Fig. 3b,e,g. As comparable methods to extract dynamical information, we compared the result of our method with those of reformulated TDMD using the Cartesian coordinates and exact DMD breaking the tensor data structure (for details, see Appendix E). In Fig. 3i, our method apparently distinguished

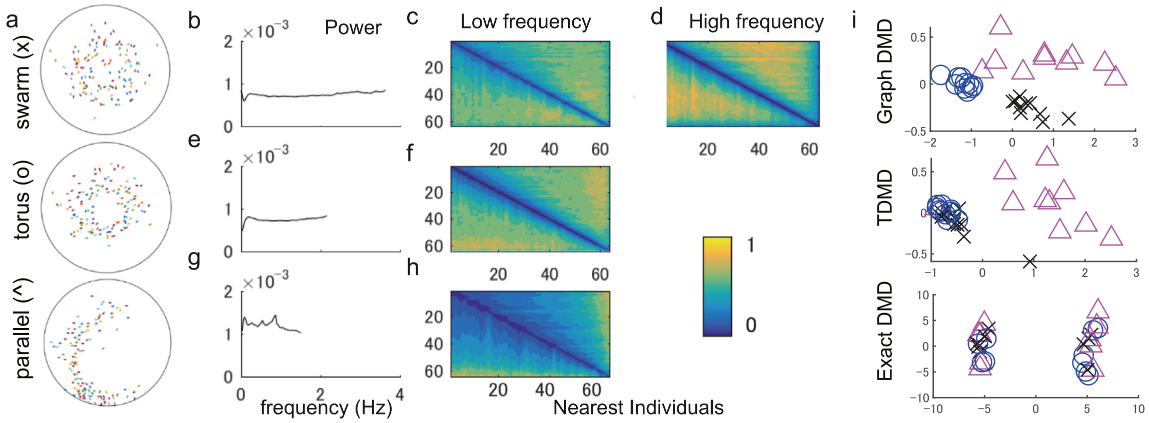

Fig. 3. Results with fish-schooling simulations. (a) Three different behavioral shapes. Temporal frequency (b,e,g) and spatial DMD spectra in low (c, f, h) and in high frequency mode (d) are shown. (i) Embedding with distance matrix of three methods. Symbols are given in (a).

the three types whereas reformulated TDMD and exact DMD did not. We quantitatively evaluated the classification error with k-nearest neighbor classifier ($k = 3$) for simplicity. We used 45 sequences in total and computed averaged 3-fold cross-validation error. The classification error in Graph DMD (0.022) was smaller than those in TDMD (0.311) and exact DMD (0.511).

8. Conclusions

In this paper, we formulated Koopman spectral analysis for NLDSs with structures among observable and proposed an estimation algorithm for performing it with a given sequence of data matrix with dependent structures among observables, which can be useful for understanding the latent global dynamics underlying such NLDSs from the available data. To this end, we first formulated the problem of estimating the spectra of Koopman operator defined in vvRKHSs to incorporate the structure among observables, and then developed a procedure for applying this to the analysis of such NLDSs by reformulating Tensor-based DMD. As a special case of our method, we proposed the method named as Graph DMD, which is a numerical algorithm for Koopman spectral analysis of graph dynamical systems, using a sequence of adjacency matrices. We further considered applications using our method, which were empirically illustrated using both synthetic and real-world datasets.

Acknowledgments

We would like to thank Naoya Takeishi, Isao Ishikawa, Masahiro Ikeda for the beneficial discussion. This work was supported by JSPS KAKENHI (Japan) Grant Numbers 16K12995, 16H01548, 18K18116, and 18H03287, and AMED (Japan) Grant Number JP18dm0307009.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.neunet.2019.04.020.

References

Alaa, A. M., & van der Schaar, M. (2017). Bayesian inference of individualized treatment effects using multi-task gaussian processes. In *Advances in neural information processing systems, Vol. 30* (pp. 3424–3432).

Álvarez, M. A., Rosasco, L., Lawrence, N. D., et al. (2012). Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, *4*(3), 195–266.

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., & Hwang, D.-U. (2006). Complex networks: Structure and dynamics. *Physics Reports*, *424*(4–5), 175–308.

Brunton, B. W., Johnson, L. A., Ojemann, J. G., & Kutz, J. N. (2016). Extracting spatial-temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. *Journal of Neuroscience Methods*, *258*, 1–15.

Bullmore, E., & Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, *10*(3), 186–198.

Caponnetto, A., Micchelli, C. A., Pontil, M., & Ying, Y. (2008). Universal multi-task kernels. *Journal of Machine Learning Research (JMLR)*, *9*(Jul), 1615–1646.

Centola, D., & Macy, M. (2007). Complex contagions and the weakness of long ties. *American Journal of Sociology* , *113*(3), 702–734.

Chung, F. R. (1997). *Spectral graph theory, Vol. 92*. American Mathematical Soc..

Cliff, O. M., Prokopenko, M., & Fitch, R. (2016). An information criterion for inferring coupling of distributed dynamical systems. *Frontiers in Robotics and AI*, *3*, 71.

Couzin, I. D., Krause, J., James, R., Ruxton, G. D., & Franks, N. R. (2002). Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, *218*(1), 1–11.

Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems, Vol. 29* (pp. 3844–3852).

Evgeniou, T., Micchelli, C. A., & Pontil, M. (2005). Learning multiple tasks with kernel methods. *Journal of Machine Learning Research (JMLR)*, *6*(Apr), 615–637.

Evgeniou, T., & Pontil, M. (2004). Regularized multi–task learning. In *ACM SIGKDD international conference on knowledge discovery and data mining (KDD'04)* (pp. 109–117). ACM.

Fujii, K., Inaba, Y., & Kawahara, Y. (2017). Koopman spectral kernels for comparing complex dynamics: Application to multiagent sport plays. In *European conference on machine learning and knowledge discovery in databases (ECML-PKDD'17)* (pp. 127–139). Springer.

Fujii, K., & Kawahara, Y. (2019). Supervised dynamic mode decomposition via multitask learning. *Pattern Recognition Letters*, *122*(1), 7–13.

Fujii, K., Kawasaki, T., Inaba, Y., & Kawahara, Y. (2018). Prediction and classification in equation-free collective motion dynamics. *PLoS Computational Biology*, *14*(11), e1006545.

Fujii, K., Takeishi, N., Kibushi, B., Kouzaki, M., & Kawahara, Y. (2019). Data-driven spectral analysis for coordinative structures in periodic systems with unknown and redundant dynamics. *bioRxiv*, 511642.

Fujii, K., Yokoyama, K., Koyama, T., Rikukawa, A., Yamada, H., & Yamamoto, Y. (2016). Resilient help to switch and overlap hierarchical subsystems in a small human group. *Scientific Reports*, *6*, http://dx.doi.org/10.1038/srep23911.

Giannakis, D., Ourmazd, A., Slawinska, J., & Zhao, Z. (2017). Spatiotemporal pattern extraction by spectral analysis of vector-valued observables, arXiv preprint arXiv:1711.02798.

Grasedyck, L., Kressner, D., & Tobler, C. (2013). A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, *36*(1), 53–78.

Heersink, B., Warren, M. A., & Hoffmann, H. (2017). Dynamic mode decomposition for interconnected control systems, arXiv preprint arXiv:1709.02883.

Hojo, M., Fujii, K., Inaba, Y., Motoyasu, Y., & Kawahara, Y. (2018). Automatically recognizing strategic cooperative behaviors in various situations of a team sport. *PLoS One*, *13*(12), e0209247.

Idé, T., & Kashima, H. (2004). Eigenspace-based anomaly detection in computer systems. In *ACM SIGKDD international conference on knowledge discovery and data mining (KDD'04)* (pp. 440–449). ACM.

Kawahara, Y. (2016). Dynamic mode decomposition with reproducing kernels for koopman spectral analysis. In *Advances in neural information processing systems, Vol. 29* (pp. 911–919).

Klus, S., Gelß, P., Peitz, S., & Schütte, C. (2018). Tensor-based dynamic mode decomposition. *Nonlinearity*, *31*(7), 3359.

Koopman, B. O. (1931). Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, *17*(5), 315–318.

Kuhlman, C. J., Kumar, V. A., Marathe, M. V., Mortveit, H. S., Swarup, S., Tuli, G., et al. (2011). A general-purpose graph dynamical system modeling framework. In *Winter simulation conference (WSC'11)* (pp. 296–308). IEEE.

Kutz, J., Fu, X., & Brunton, S. L. (2016). Multiresolution dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, *15*(2), 713–735.

Liu, W., Zha, Z.-J., Wang, Y., Lu, K., & Tao, D. (2016). *p*-Laplacian Regularized sparse coding for human activity recognition. *IEEE Transactions on Industrial Electronics*, *63*(8), 5120–5129.

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

Micchelli, C. A., & Pontil, M. (2005a). Kernels for multi–task learning. In *Advances in neural information processing systems, Vol. 17* (pp. 921–928).

Micchelli, C. A., & Pontil, M. (2005b). On learning vector-valued functions. *Neural Computation*, *17*(1), 177–204.

Mortveit, H. S., & Reidys, C. M. (2001). Discrete, sequential dynamical systems. *Discrete Mathematics*, *226*(1–3), 281–295.

Mortveit, H., & Reidys, C. (2007). *An introduction to sequential dynamical systems*. Springer Science & Business Media.

Oseledets, I. V. (2011). Tensor-train decomposition. *SIAM Journal on Scientific Computing*, *33*(5), 2295–2317.

Proctor, J. L., Brunton, S. L., & Kutz, J. (2016). Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, *15*(1), 142–161.

Proctor, J. L., & Eckhoff, P. A. (2015). Discovering dynamic patterns from infectious disease data using dynamic mode decomposition. *International Health*, *7*(2), 139–145.

Quang, M. H., Kang, S. H., & Le, T. M. (2010). Image and video colorization using vector-valued reproducing kernel Hilbert spaces. *Journal of Mathematical Imaging and Vision*, *37*(1), 49–65.

Rowley, C. W., Mezić, I., Bagheri, S., Schlatter, P., & Henningson, D. S. (2009). Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, *641*, 115–127.

Schmid, P. J. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, *656*, 5–28.

Schrödl, S. J. (2009). *Operator-valued reproducing kernels and their application in approximation and statistical learning* (Ph.D. thesis), Technische Universität München.

Seo, Y., Defferrard, M., Vandergheynst, P., & Bresson, X. (2017). Structured sequence modeling with graph convolutional recurrent networks, arXiv preprint arXiv:1612.07659.

Susuki, Y., & Mezić, I. (2011). Nonlinear koopman modes and coherency identification of coupled swing dynamics. *IEEE Transactions on Power Systems*, *26*(4), 1894–1904.

Susuki, Y., & Mezić, I. (2014). Nonlinear Koopman modes and power system stability assessment without models. *IEEE Transactions on Power Systems*, *29*(2), 899–907.

Takeishi, N., Kawahara, Y., Tabei, Y., & Yairi, T. (2017). Bayesian dynamic mode decomposition. In *International joint conference on artificial intelligence (IJCAI'17)* (pp. 2814–2821).

Takeishi, N., Kawahara, Y., & Yairi, T. (2017). Learning koopman invariant subspaces for dynamic mode decomposition. In *Advances in neural information processing systems, Vol. 30* (pp. 1130–1140).

Takeishi, N., Kawahara, Y., & Yairi, T. (2017a). Sparse nonnegative dynamic mode decomposition. In *2017 IEEE international conference on image processing (ICIP'17)* (pp. 2682–2686). IEEE.

Takeishi, N., Kawahara, Y., & Yairi, T. (2017b). Subspace dynamic mode decomposition for stochastic Koopman analysis. *Physical Review E*, *96*, 033310.

Takeuchi, K., Kawahara, Y., & Iwata, T. (2017). Structurally regularized non-negative tensor factorization for spatio-temporal pattern discoveries. In *European conference on machine learning and knowledge discovery in databases (ECML-PKDD'17)* (pp. 582–598). Springer.

Taubin, G. (1995). A signal processing approach to fair surface design. In *International conference on computer graphics and interactive techniques (SIGGRAPH'95)* (pp. 351–358). ACM.

Tu, J. H., Rowley, C. W., Luchtenburg, D. M., Brunton, S. L., & Kutz, J. N. (2014). On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, *1*(2), 391–421.

Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., & Shochet, O. (1995). Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, *75*(6), 1226–1229.

Williams, M. O., Kevrekidis, I. G., & Rowley, C. W. (2015). A data-driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, *25*(6), 1307–1346.

Wu, C. W. (2005). Synchronization in networks of nonlinear dynamical systems coupled via a directed graph. *Nonlinearity*, *18*(3), 1057–1064.