Probabilistic Models Characterized by a Kernel Matrix and Their Learning Methods

by

Takahiro Kawashima

Dissertation

submitted to the Department of Statistical Science

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

The Graduate University for Advanced Studies, SOKENDAI

March 2024

Acknowledgments

First and foremost, I would like to express my utmost gratitude to Prof. Hideitsu Hino (ISM), my doctoral supervisor. When I entered the master's program at the University of Electro-Communications (UEC), I was unsure about my research direction. However, Prof. Hino provided me with a highly interesting direction and continued to discuss with his extensive knowledge. After enrolling in the Ph.D. program, he supported my research life even more strongly than before. While guiding me in his role as an educator, I feel that he also acknowledged me as a collaborative researcher. For instance, when reviewing drafts of my papers, he suggested improvements in nuanced details while provided constructive comments whenever my scattered ideas needed shaping. Furthermore, he is someone I deeply respect both as a researcher and an educator, and I have learned numerous things from him beyond the direct guidance. He also supported me financially since my master's program, and I attribute this dissertation to Prof. Hino's unwavering support. I may not have been an outstanding student, but I sincerely hope that this dissertation and research works can serve as a modest way to repay Prof. Hino for his continuous support.

I would also like to express my gratitude to the members of the thesis examination committee: Prof. Kengo Kamatani (ISM) , Assoc. Prof. Daichi Mochihashi (ISM), and Lect. Naoya Takeishi (UTokyo). Prof. Kamatani carefully reviewed this doctoral dissertation, identifying several crucial points for improvement. Assoc. Prof. Mochihashi, drawing upon profound knowledge and experience, pointed out multiple discussions necessary for enhancing this dissertation. Lect. Takeishi took charge of reviewing our paper for Neural Computation, which corresponds to Chapter 3 of this thesis, and also provided insightful comments during the thesis examination.

When I transferred from a KOSEN (college of technology) to UEC, I initially chose a department with the intention of pursuing studies in a field other than machine learning. However, after a brief period of wandering, I developed an interest in machine learning. It was truly fortunate for me that Prof. Hayaru Shouno (UEC) had been directing a lab dedicated to machine learning in that department. For my undergraduate research, Prof. Shouno provided me with a progressive yet educational theme, and in the master's program, he warmly supported me as I ventured to other research institutions, allowing me the freedom to explore my interests. I am sincerely grateful for the guidance and support from Prof. Shouno.

During my graduate programs, I concurrently worked as a part-time biostatistician at National Center of Neurology and Psychiatry, Japan (NCNP). This experience not only provided financial support but also allowed me to engage comprehensively in projects, from consulting on data analysis and planning to paper publication. It was an invaluable opportunity that enriched my skills. I am sincerely grateful to Prof. Hisateru Tachimori (Keio) at NCNP for accepting me into the team. The five years I spent working in a different field at NCNP were made possible, primarily due to Prof. Tachimori's excellent managerial skills and the depth of understanding he had for me, coming from an engineering background. Furthermore, I would like to thank my former colleague in NCNP, Asst. Prof. Yui Tomo (Keio) for advancing data analysis together on several projects.

I am grateful to the members of Hino lab and my peers at SOKENDAI for their support not only in studies but also in the daily life as a student. In particular, I appreciate the advice and support from Mr. Masanari Kimura during my job hunting process, and Mr. Keishi Sando played a crucial role in revitalizing the student room, which had become quiet amid and after the COVID-19 pandemic.

Finally, I express my gratitude to everyone I have encountered throughout my research journey. Thank you to each and every individual who has been a part of my academic and personal growth.

Abstract

Positive definite kernels play a significant role in modern machine learning. Kernel methods opened up possibilities for analyzing complex data that may be governed by nonlinear structure because of the rich representational power and nice theoretical properties. Intuitively a positive definite kernel realizes nonlinear data processing on the input space in which data points are located by determining a metric on a possibly infinite-dimensional space. Related to the kernel methods, probabilistic models characterized by a positive definite kernel have also attracted attention. Gaussian processes (GPs) and determinantal point processes (DPPs) fall into these models. These models offer choices of methodologies for dealing with complex data as well as the kernel methods. This dissertation addresses the following two topics about probabilistic models with a positive definite kernel:

(i) We propose a GP-based generative model for multivariate time-series data via a physics approach and developing an efficient inference method for the model.

(ii) We develop a simple and fast learning method for DPPs.

For (i), we propose a nonlinear and probabilistic generative model of Koopman mode decomposition (KMD) based on the framework of unsupervised GPs. Differential equations appear in many fields of science, including materials science, geophysics, epidemiology, and social informatics. In these fields, multivariate time-series data governed by an unknown differential equation is sometimes obtained, and we may want to know about the underlying dynamics. One of the factors that make the estimation problem of the underlying dynamics difficult is nonlinearity. Through KMD, nonlinear dynamics on a finite-dimensional space is lifted into an infinite-dimensional space in which the dynamics behaves linearly. That leads concrete algorithms to find the modes

characterizing the dynamics, such as dynamic mode decomposition (DMD). While DMD and other related algorithms have been successful in many fields, resulting values yielded by the algorithms are sometimes not very easy-to-interpret. On the other hand, our model makes it possible to estimate the physical quantities associated with the Koopman modes and the (low-dimensional) latent variables simultaneously by taking an approach of generative modeling. Our model is the first to give a way to estimate KMD latent variables, and we show the usefulness through some numerical experiments with both of synthetic and real-world datasets. Moreover, we develop a scheme that reduces the computational complexity to learn our model for scalability.

For (ii), we develop a fast, stable, and simple learning rule for DPPs on the basis of MM (minorization-maximization) algorithms, which increases the objective values monotonically. DPPs are powerful probabilistic models that generate random subsets with diverse items from a ground set. For example, let us consider a recommender system on an e-commerce site in which a variety of home appliances are handled. Then, the purchasing histories can be regarded as samples of a DPP on the finite ground set, which consists of all the products handled by the site. Now, we may want to assume that "rarely do consumers buy more than one refrigerator at a time," or more conceptually: "a random subset tends not to have similar items simultaneously." Such a concept is called negative dependence, and DPPs take it into account. Since the similarities between items are parameterized as a kernel matrix in DPPs on a finite set, the fitting problem of DPPs becomes a problem of estimating a positive definite matrix. Although some existing studies have addressed the problem, there is room for improving the stability and speed of convergence. In this work, we show that the learning problem of DPPs can be resulted in iterative solving of a continuous algebraic Riccati equation (CARE), which is a solvable class of quadratic matrix equations. The monotonicity of our algorithm follows the property of MM algorithms. We also develop an acceleration technique for our algorithm by introducing a step size parameter whose value can be determined adaptively in each iteration. We numerically compare our algorithm and existing methods with synthetic and real-world data in experiments. Our algorithm outperforms existing methods in convergence speed for most of the datasets, and we additionally discuss what contributes the efficiency of our algorithm.

This dissertation is organized as follows. In Chapter 1, we give an introduction motivating us to study probabilistic models parameterized by a kernel matrix. In Chapter

2, we present technical preliminaries related to kernel methods, GPs, and DPPs. In Chapter 3, we develop GPKMD based on Bayesian DMD. We show that Bayesian DMD can be extended to a GP-based probabilistic generative model naturally. We also propose a computational scheme to improve the time complexity of learning GPKMD. Experimental results find that GPKMD can capture important dynamics from observed data. In Chapter 4, we study an efficient and simple rule to learn full-rank DPPs. We prove that maximum likelihood estimation of a DPP can be reduced to iterative solving of some matrix quadratic equation by using MM algorithm. We also develop an accelerated version of the algorithm which is no longer monotone increasing but possibly converges faster. Numerical results on both synthetic and real-world datasets show our algorithm outperforms existing methods. Finally, we give concluding remarks in Chapter 5.

Contents

List of Figures

List of Tables

1 Introduction

The remarkable and rapid progress of information technology in recent years is beyond dispute. One crucial factor contributing to this progress is the concept of "data accumulation," which has become a pillar alongside the evolution of hardware and infrastructures and the sophistication of software for information processing over the past quarter century. As a result, technological possibilities have taken a significant leap forward. For instance, being stored diverse data, including multivariate time-series histories for weather forecasting, purchase histories on e-commerce sites, and videos, images, and texts shared on social media, a mathematical foundation for extracting non-trivial knowledge from such complex-structured data becomes necessary. This social background has driven the development of machine learning methodologies that go beyond classical statistics.

What kind of data are difficult to handle using classical statistics? Now consider the following three sentences:

- Seeing an aquarium is one of my hobbies.

- I would like to see aquatic life.
- Tokyo is the greatest prefecture in Japan.

When we imagine a hypothetical space in which pairs of sentences with similar meanings are placed closer together, it is guessed that "Seeing an aquarium is one of my hobbies" and "I would like to see aquatic life" are closely located, while "Tokyo is the greatest prefecture in Japan" is farther away from them. And we can have an intuition that a difficulty lies in determining an appropriate origin in the space, namely, treating natural languages as vectors (i.e., elements of a vector space) may not be appropriate. This is one of the reasons that classical statistics is not applicable to modern complex data. On the other hand, many machine learning algorithms essentially depend on only pairwise similarities between data points. Even if an appropriate origin is undetermined, quantifying similarities can suffice. In the example above, we may know that "Seeing an aquarium is one of my hobbies" is similar to "I would like to see aquatic life," but dissimilar to "Tokyo is the greatest prefecture in Japan."

Methods based on kernel functions explicitly leverage this concept of similarity. A kernel function maps a pair of data points to the non-negative similarity value. Its shape can be specified flexibly according to the structure of the problem. To be precise, this is equivalent to determining the reproducing kernel Hilbert space (RKHS) which associates with the feature map of the data points. The Gaussian kernel

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\ell}\right),$$

where $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ are the inputs in the input space $\mathcal{X}$ and $\ell > 0$ is the hyperparameter, is one of the most representative kernel functions. In the Gaussian kernel, the data points $\boldsymbol{x}$ and $\boldsymbol{x}'$ contribute to the value solely through their difference $\boldsymbol{x} - \boldsymbol{x}'$, implying that the input space $\mathcal{X}$ can be an affine space, which has no origin. Furthermore, by choosing such a kernel which induces a nonlinear metric, we can handle nonlinear-structured data with taking the higher moments into account. This is because by "kernelizing" linear methods, data points are mapped to an infinite-dimensional feature space and then the linear processing is applied in the feature space.

Methods associated with a kernel function are called kernel methods, and they have been used to nonlinearize many machine learning algorithms since around 2000. Kernel methods find application in various problem types including supervised/unsupervised learning and classification/regression like support vector machines [Cortes and Vapnik, 1995], ridge regression [Shawe-Taylor and Cristianini, 2004, Chapter 7], and principal component analysis [Schölkopf et al., 1997]. Nowadays the scope of kernel methods extends beyond primitive extensions of linear algebraic techniques. Nonparametric statistical tests of independence [Gretton et al., 2005], two-sample tests [Gretton et al., 2012], and nonparametric variational inference [Liu and Wang, 2016] are the excellent examples that exploit good mathematical properties of RKHSs.

Gaussian processes (GPs) are one of the probabilistic models characterized by kernel functions. GPs are nonparametric stochastic processes on a function space and can be seen as the probabilistic models of kernel methods in a sense. GPs can be, therefore, seemlessly integrated into probabilistic modeling frameworks. While functions following GPs take the value range $(-\infty, \infty)$ in general, they can be flexibly incorporated into probabilistic models in various ways:

- predict probabilities within $(0, 1)$ through a logistic likelihood [Rasmussen and Williams, 2008, Chapter 3]
- model intensity functions in Poisson point processes with exponential transformation which ensures non-negative values [Samo and Roberts, 2015].

Another notable advantage of GPs lies in their ability to quantify prediction uncertainty, supporting many applications in engineering such as Bayesian optimization [Shahriari et al., 2016].

Determinantal point processes (DPPs) are alternative probabilistic models characterized by a kernel function. A DPP generates random subsets from a universal set, and the kernel function controls how the items co-occur within the subsets. Originating in the field of statistical physics, DPPs have been introduced to describe the probabilistic behavior of fermions in [Macchi, 1975]. While they initially piqued the interest of mathematical physicists and probability mathematicians, DPPs later caught the attention of mathematical engineers due to their attractive property called negative dependence, which makes the random subsets tending not to have similar items simultaneously. Now DPPs have gained attention in many application fields, including the machine learning community inspired by the excellent review article [Kulesza and Taskar, 2012].

Based on the background, we develop probabilistic methods stand on a kernel matrix. As complexity of data increases, machine learning methods need to be more interpretable. In this sense, probabilistic modeling has an advantage over deterministic kernel methods. Since we usually build probabilistic models based on some prior knowledge, the learning results may provide meaningful insight about the data. In particular, this dissertation addresses the following two topics:

- We propose a GP-based generative model for multivariate time-series data via a physics approach and an efficient inference method for the model.
- We develop a simple and fast learning method for DPPs.

We discuss these two topics based on the published papers [Kawashima and Hino, 2022, 2023], after presenting a technical introduction about kernel methods, GPs, and DPPs in Chapter 2.

We propose a probabilistic generative model of multivariate time-series data with an unsupervised GP in Chapter 3. Triggered by neural ODEs [Chen et al., 2018], many researchers are addressing data-driven identification problems of unknown differential equations, and these studies continue to evolve as part of physics-informed machine learning (PIML) [Karniadakis et al., 2021]. Because differential equations and related complex data appear in various fields of science, PIML methods have a wide range of applications, including plasma physics [Mathews et al., 2021], materials science [Lu et al., 2020], and geophysics [Zhu et al., 2021]. In particular, an approach known as operator-theoretic data analysis aims to describe data generation processes dependent on unknown differential equations using an operator such as a Koopman operator [Mezić, 2005, Rowley et al., 2009, Schmid, 2010]. This approach transforms the problem of finding a solution to the differential equation into a task of estimating quantities characterizing the operator. Operator-theoretic data analysis starts with viewing the nonlinear time evolution on a finite-dimensional space as a linear time evolution on an infinite-dimensional space. This concept resembles the kernel methods that perform nonlinear data processing by lifting the finite-dimensional data into an infinite-dimensional RKHS. In fact, Kostic et al. [2022] have found a kind of duality between Koopman operator regression and conditional kernel embedding. Inspired by this similarity, we propose a nonlinear and probabilistic generative model for operator-theoretic data analysis based on GPs.

In Chapter 4, we develop an inference method for DPPs on a finite ground set on the basis of MM (minorization-maximization) algorithms [Hunter and Lange, 2004, Sun et al., 2017]. DPPs on a finite ground set are generally parameterized by a positive semidefinite kernel matrix (Gram matrix) consists of pairwise similarities of the items. From this tractable property, a finite ground set is likely to be considered while DPPs can also be defined on an infinite set (possibly uncountable). Because DPPs have negative dependence, by which dissimilar items tend to be yielded in random subsets, such an approach can be applied to recommender systems for example; we may assume "rarely do consumers buy more than one refrigerator at a time." Currently the learning problem of DPPs arises in somewhat limited applications such as recommendation [Gillenwater et al., 2014], image search [Kulesza and Taskar, 2011], and document summarization [Dupuy and Bach, 2018], but it is expected to have wider applications as DPPs become more widespread in the machine learning community. Although some structure is often assumed to a kernel matrix for reducing the computational cost [Gartrell et al., 2017, Mariet and Sra, 2016, Dupuy and Bach, 2018], it may be preferred to give no specific structure if we have no prior knowledge. We thus develop an algorithm for learning DPPs without specific structure for the kernel matrix.

Chapter 5 concludes this dissertation. While our methods developed in this dissertation show certain levels of effectiveness with numerical experiments, there is still future work to be considered. The future directions of our study are also discussed in Chapter 5.

2
Preliminary

In this dissertation, we consistently consider probabilistic models characterized by a kernel matrix. In this section, we introduce kernel methods and probabilistic models associated with a kernel: Gaussian processes and determinantal point processes.

2.1 Kernel Methods

2.1.1 From Linear to Nonlinear: an Example on Kernel Ridge Regression

We start from finite-dimensional multivariate analysis with an example on ridge regression. Let $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ be the data consisting of N pairs of points in $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^D$ is the input space (including intercepts) and $\mathcal{Y} \subseteq \mathbb{R}$ is the output space. Ridge regression learns a linear function $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$ by solving

$$\min_{\boldsymbol{w} \in \mathbb{R}^D} \sum_{i=1}^{N} (y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)^2 + \lambda \|\boldsymbol{w}\|^2, \tag{2.1}$$

where $\lambda \geq 0$ is the regularization parameter which controls the smoothness of f. The problem (2.1) has an analytic solution. With the input matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)^\top \in \mathbb{R}^{N \times D}$ and the output vector $\boldsymbol{y} = (y_1, \ldots, y_N)^\top \in \mathbb{R}^N$, the solution is

$$\boldsymbol{w} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top \boldsymbol{y} = \boldsymbol{X}^\top (\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I})^{-1} \boldsymbol{y}, \tag{2.2}$$

where the second equality holds from the following identity (see Lemma 2 in [Welling, 2010])

$$(\boldsymbol{P}^{-1} + \boldsymbol{B}^\top \boldsymbol{R}^{-1} \boldsymbol{B}) \boldsymbol{B}^\top \boldsymbol{R}^{-1} = \boldsymbol{P}\boldsymbol{B}^\top (\boldsymbol{B}\boldsymbol{P}\boldsymbol{B}^\top + \boldsymbol{R})^{-1}.$$

By plugging-in the solution (2.2), we can predict an output value y_* corresponding to a new input $\boldsymbol{x}_*$ as

$$y_* = \boldsymbol{w}^\top \boldsymbol{x}_* = \boldsymbol{y}^\top (\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I})^{-1} \boldsymbol{X} \boldsymbol{x}_*. \tag{2.3}$$

Since the prediction (2.3) depends only on the Euclidean inner products of the input points, it can be rewritten as

$$y_* = \boldsymbol{y}^\top (\boldsymbol{K} + \lambda \boldsymbol{I})^{-1} \boldsymbol{k}_*, \tag{2.4}$$

where

$$\boldsymbol{K} = \begin{pmatrix} \langle \boldsymbol{x}_1, \boldsymbol{x}_1 \rangle & \langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle & \cdots & \langle \boldsymbol{x}_1, \boldsymbol{x}_N \rangle \\ \langle \boldsymbol{x}_2, \boldsymbol{x}_1 \rangle & \langle \boldsymbol{x}_2, \boldsymbol{x}_2 \rangle & \cdots & \langle \boldsymbol{x}_2, \boldsymbol{x}_N \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \boldsymbol{x}_N, \boldsymbol{x}_1 \rangle & \langle \boldsymbol{x}_N, \boldsymbol{x}_2 \rangle & \cdots & \langle \boldsymbol{x}_N, \boldsymbol{x}_N \rangle \end{pmatrix}, \tag{2.5}$$

$$\boldsymbol{k}_* = (\langle \boldsymbol{x}_1, \boldsymbol{x}_* \rangle, \langle \boldsymbol{x}_2, \boldsymbol{x}_* \rangle, \ldots, \langle \boldsymbol{x}_N, \boldsymbol{x}_* \rangle)^\top, \tag{2.6}$$

with the Euclidean inner product $\langle \cdot, \cdot \rangle$.

The inner product view of the ridge regression (2.3) motivates us to consider its nonlinear generalization; kernel ridge regression uses a generally nonlinear similarity measure $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, which is called the kernel function, in (2.5) and (2.6). By choosing an appropriate kernel function and the regularization parameter λ, the kernel ridge regression produces a good nonlinear predictor $f : \mathcal{X} \to \mathcal{Y}$.

2.1.2 Fundamentals of Kernel Methods

In the previous subsection, we saw that the ridge regression is naturally generalized to be nonlinear by the "kernelization." Many other linear machine learning methods can also be kernelized in similar manners. However, we need to introduce some mathematical preparations to understand why, when, and how the kernel methods work.

A kernel method does not work with any kernel function. As seeing later, a kernel function satisfying the following properties induces an appropriate inner product space.

Definition 2.1 (Positive definite kernel)**.** *A kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive definite if $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}', \boldsymbol{x})$ for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ and*

$$\sum_{i,j=1}^{N} c_i c_j^* k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$$

for any $N \in \mathbb{N}$, $c_1, \dots, c_N \in \mathbb{C}$, and $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N \in \mathcal{X}$, where c^* denotes the complex conjugate of c.

Definition 2.1 is equivalent to the kernel matrix (or Gram matrix)

$$\boldsymbol{K} = \begin{pmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & k(\boldsymbol{x}_1, \boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ k(\boldsymbol{x}_2, \boldsymbol{x}_1) & k(\boldsymbol{x}_2, \boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_2, \boldsymbol{x}_N) \\ \vdots & \vdots & \ddots & \ddots \\ k(\boldsymbol{x}_N, \boldsymbol{x}_1) & k(\boldsymbol{x}_N, \boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{pmatrix} \tag{2.7}$$

being positive (semi-) definite for any $N \in \mathbb{N}$ and $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N \in \mathcal{X}$. Given $\boldsymbol{x} \in \mathcal{X}$, we denote $k(\cdot, \boldsymbol{x})$ as the function on $\mathcal{X}$ with the fixed second argument: $k(\cdot, \boldsymbol{x}) : \boldsymbol{x}' \mapsto k(\boldsymbol{x}', \boldsymbol{x})$.

An inner product space $\mathcal{H}$ is called a Hilbert space if it is complete, that is, every

Cauchy sequence in $\mathcal{H}$ has its limit also in $\mathcal{H}$. In particular, the following reproducing kernel Hilbert spaces are the stages on which kernel methods are based.

Definition 2.2 (Reproducing kernel Hilbert space [Saitoh and Sawano, 2016])**.** *Let $\mathcal{H}$ be a Hilbert space consisting of functions whose input space is $\mathcal{X}$ and denote the inner product on $\mathcal{H}$ as $\langle f, g\rangle_{\mathcal{H}}$ for $f, g \in \mathcal{H}$. If there exists $k(\cdot, \boldsymbol{x}) \in \mathcal{H}$ for any $\boldsymbol{x} \in \mathcal{X}$ that satisfies*

$$\langle f, k(\cdot, \boldsymbol{x})\rangle_{\mathcal{H}} = f(\boldsymbol{x})$$

for an arbitrary $f \in \mathcal{H}$, $\mathcal{H}$ is called the reproducing kernel Hilbert space (RKHS), and k is called the reproducing kernel.

The following theorem states about the indivisible connection between a positive definite kernel and an RKHS.

Theorem 2.3 (Moore–Aronszajn [Aronszajn, 1950, Berlinet and Thomas-Agnan, 2004, Theorem 3])**.** *For a positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, there exists a unique RKHS $\mathcal{H}$ of functions on $\mathcal{X}$ with the reproducing kernel k.*

Conversely, the reproducing kernel k of an RKHS $\mathcal{H}$ is also a positive definite kernel since

$$\sum_{i,j=1}^{N} c_i c_j^* k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left\langle \sum_{i=1}^{N} c_i k(\cdot, \boldsymbol{x}_i), \sum_{j=1}^{N} c_j k(\cdot, \boldsymbol{x}_j) \right\rangle_{\mathcal{H}} = \left\| \sum_{i=1}^{N} c_i k(\cdot, \boldsymbol{x}_i) \right\|_{\mathcal{H}}^2 \geq 0$$

holds for any $N \geq 0$, $c_1, \ldots, c_N \in \mathbb{C}$, and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathcal{X}$, where $\|f\|_{\mathcal{H}}$ is the norm of $f \in \mathcal{H}$ defined as $\|f\|_{\mathcal{H}} = \sqrt{\langle f, f\rangle_{\mathcal{H}}}$ These results states that we can equate positive definite kernels and reproducing kernels.

Let us define a map $\boldsymbol{\phi} : \mathcal{X} \ni \boldsymbol{x} \mapsto k(\cdot, \boldsymbol{x}) \in \mathcal{H}$. Given a positive definite kernel k, the value at $(\boldsymbol{x}, \boldsymbol{x}')$ is

$$k(\boldsymbol{x}, \boldsymbol{x}') = \langle k(\cdot, \boldsymbol{x}), k(\cdot, \boldsymbol{x}')\rangle_{\mathcal{H}} = \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x}')\rangle_{\mathcal{H}} \,. \tag{2.8}$$

In general, an inner product is said to give a similarity measure on the space, since the value takes zero iff the parameters are orthogonal and becomes larger if the parameters point in similar directions. The value of the kernel function (2.8), therefore, can be regarded as the similarity between $\boldsymbol{x}$ and $\boldsymbol{x}'$ on the high-dimensional space $\mathcal{H}$. In this sense, $\boldsymbol{\phi}$ and $\mathcal{H}$ are often called a feature map and a feature space, respectively. If the representation power of $k(\cdot, \boldsymbol{x})$ is enough, kernel methods using the kernel k may perform well even for complex and nonlinear data.

2.1.3 Nyström Approximation

As we see in Subsection 2.1.1, kernel methods based on a positive definite kernel k depend on the $N \times N$ kernel matrix $\boldsymbol{K}$ (defined in (2.7)) and involve some $O(N^3)$ operations such as matrix inverses. This implies that kernel methods do not scale for problems with large samples. One conceivable approach to address this problem is low-rank approximation of $\boldsymbol{K}$. For example, if we have $\boldsymbol{R} \in \mathbb{R}^{N \times P}$, $\mathbb{N} \ni P < N$ such that $\boldsymbol{K} \approx \boldsymbol{R}\boldsymbol{R}^\top$, the inverse of $\boldsymbol{K} + \lambda \boldsymbol{I}$ in (2.3) is approximated by

$$(\boldsymbol{K} + \lambda\boldsymbol{I})^{-1} \approx (\boldsymbol{R}\boldsymbol{R}^\top + \lambda\boldsymbol{I})^{-1} = \lambda^{-1}(\boldsymbol{I} - \boldsymbol{R}(\boldsymbol{R}^\top\boldsymbol{R} + \lambda\boldsymbol{I})^{-1}\boldsymbol{R}^\top)$$

and the computational cost reduces down to $O(N^2P)$ (and $O(NP^2)$ for approximating $(\boldsymbol{K} + \lambda\boldsymbol{I})^{-1}\boldsymbol{y}$).

One favorable approach to get $\boldsymbol{R}$ is Nyström approximation. The Nyström approximation was first introduced in [Williams and Seeger, 2000], and its variation and theoretical analysis were proposed in [Drineas and Mahoney, 2005]. Drineas and Mahoney [2005] showed that we can obtain a rank-P approximation of $\boldsymbol{K} \in \mathbb{S}_+^N$ by:

1. Choose $\tilde{N} \in \mathbb{N}$ such that $P \leq \tilde{N} \leq N$.

2. Compute $p_i = K_{ii}^2 / \sum_{j=1}^N K_{jj}^2$ for $i = 1, \ldots, N$.

3. Sample $\tilde{N}$ times from the set $\{1, \ldots, N\}$ with replacement and with respect to the probabilities $\{p_1, \ldots, p_N\}$, and let $\mathcal{I}$ be the set of the sampled indices[1].

4. Assign $\boldsymbol{C} = (K_{ij})_{i \in \{1,\ldots,N\}, j \in \mathcal{I}} \in \mathbb{R}^{N \times \tilde{N}}$, $\boldsymbol{W} = (K_{ij})_{i,j \in \mathcal{I}} \in \mathbb{R}^{\tilde{N} \times \tilde{N}}$, and $\boldsymbol{D} = \mathrm{diag}(\{(\tilde{N}p_i)^{-\frac{1}{2}}\}_{i \in \mathcal{I}})$.

5. Obtain $\boldsymbol{K} \approx \tilde{\boldsymbol{K}}_P := \boldsymbol{C}\boldsymbol{W}_P^+\boldsymbol{C}^\top$, where $\boldsymbol{W}_P$ denotes the best rank-P approximation of $\boldsymbol{D}\boldsymbol{W}\boldsymbol{D}$ and $\boldsymbol{W}_P^+$ is its pseudoinverse.

[1] We allow duplication of elements within the set $\mathcal{I}$. Thus $\mathcal{I}$ is precisely a tuple.

The following theorem justifies the Nyström approximation.

Theorem 2.4 (Drineas and Mahoney [2005], Theorem 3)**.** *Let $\varepsilon > 0, \delta \in (0,1), \eta = 1 + \sqrt{8\log(1/\delta)}$, and $\boldsymbol{K}_P$ be the best rank-P approximation to $\boldsymbol{K}$. If $\tilde{N} \geq 64P/\varepsilon^4$, then*

$$\mathbb{E}[\|\boldsymbol{K} - \tilde{\boldsymbol{K}}_P\|_F] \leq \|\boldsymbol{K} - \boldsymbol{K}_P\|_F + \varepsilon \sum_{n=1}^{N} K_{nn}^2.$$

And if $\tilde{N} \geq 64\eta^2 P/\varepsilon^4$, then

$$\mathbb{P}\left(\|\boldsymbol{K} - \tilde{\boldsymbol{K}}_P\|_F \leq \|\boldsymbol{K} - \boldsymbol{K}_P\|_F + \varepsilon \sum_{n=1}^{N} K_{nn}^2\right) \geq 1 - \delta$$

also holds.

Note that a similar result has also shown in the sense of the spectral norm in [Drineas and Mahoney, 2005]. The advantage of Nyström approximation is that we need not to compute and store the large matrix $\boldsymbol{K}$. Once the sampling probabilities $\{p_i\}_{i=1}^N$ are computed from the diagonal elements $\{K_{ii}\}_{i=1}^N$ and the set of the sampled indices $\mathcal{I}$ is obtained, our requirements are the submatrices of $\boldsymbol{K}$: $\boldsymbol{C} \in \mathbb{R}^{N \times \tilde{N}}$ and $\boldsymbol{W} \in \mathbb{R}^{\tilde{N} \times \tilde{N}}$. This leads to a remarkable reduction in both computational and space complexity. The computational complexity of the Nyström approximation is $O(\tilde{N}^3 + N\tilde{N}P)$ and the space complexity is also reduced to $O(\tilde{N}N)$ from $O(N^2)$.

2.2 Gaussian Processes

A Gaussian process (GP) is an infinite-dimensional distribution on a function space. GPs are often understood to be probabilistic variants of kernel methods. This subsection gives a brief introduction to GPs.

2.2.1 From Bayesian Linear Regression to GP Regression

Consider a Bayesian linear regression problem from the input space $\mathcal{X} \subseteq \mathbb{R}^D$ to the output space $\mathcal{Y} \subseteq \mathbb{R}$ with given observations $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, where $x_{i1} = 1$ for $i = 1, \ldots, N$.

The Bayesian linear regression problem is to find the posterior $p(\boldsymbol{w}|\{\boldsymbol{x}_i\}_{i=1}^N, \boldsymbol{y})$ from the following Bayesian model:

$$\begin{aligned} &p(y_i|\boldsymbol{x}_i, \boldsymbol{w}; \sigma^2) = \mathcal{N}(\boldsymbol{w}^\top \boldsymbol{x}_i, \sigma^2) \ \text{ for } i = 1, \ldots, N, \\ &p(\boldsymbol{w}; \sigma_w^2) = \mathcal{N}(\boldsymbol{0}_D, \sigma_w^2 \boldsymbol{I}_D). \end{aligned}$$

Denoting $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)^T$, the posterior is yielded analytically as

$$p(\boldsymbol{w}|\{\boldsymbol{x}_i\}_{i=1}^N, \boldsymbol{y}) = \mathcal{N}(\Sigma \boldsymbol{X}^\top \boldsymbol{y}, \sigma^2 \Sigma), \tag{2.9}$$

where $\Sigma = (\boldsymbol{X}^\top \boldsymbol{X} + \frac{\sigma^2}{\sigma_w^2} \boldsymbol{I})^{-1}$. Note that the mean vector in (2.9) has the same form as the solution of the ridge regression (2.2) and the noise ratio $\frac{\sigma^2}{\sigma_w^2}$ is corresponding to the regularization parameter λ. The marginal likelihood is also obtained analytically:

$$\begin{aligned} p(\boldsymbol{y}|\{x_i\}_{i=1}^N; \sigma^2) &= \int p(\boldsymbol{y}|\{x_i\}_{i=1}^N, \boldsymbol{w}; \sigma^2) p(\boldsymbol{w}; \sigma_w^2) d\boldsymbol{w} \\ &= \mathcal{N}(\boldsymbol{0}, \sigma_w^2 \boldsymbol{X}\boldsymbol{X}^\top + \sigma^2 \boldsymbol{I}) \\ &= \mathcal{N}(\boldsymbol{0}, \boldsymbol{K} + \sigma^2 \boldsymbol{I}), \end{aligned} \tag{2.10}$$

where

$$\boldsymbol{K} = \sigma_w^2 \begin{pmatrix} \langle \boldsymbol{x}_1, \boldsymbol{x}_1 \rangle & \langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle & \cdots & \langle \boldsymbol{x}_1, \boldsymbol{x}_N \rangle \\ \langle \boldsymbol{x}_2, \boldsymbol{x}_1 \rangle & \langle \boldsymbol{x}_2, \boldsymbol{x}_2 \rangle & \cdots & \langle \boldsymbol{x}_2, \boldsymbol{x}_N \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \boldsymbol{x}_N, \boldsymbol{x}_1 \rangle & \langle \boldsymbol{x}_N, \boldsymbol{x}_2 \rangle & \cdots & \langle \boldsymbol{x}_N, \boldsymbol{x}_N \rangle \end{pmatrix} \tag{2.11}$$

is the Gram matrix with the Euclidean inner product. GP regression replaces the inner product in the Gram matrix (2.11) with a positive definite kernel $k(\cdot, \cdot)$ to capture nonlinear relationship between $\boldsymbol{x}_i$ and y_i (the positive coefficient $\sigma_w^2 > 0$ can be incorporated into the kernel k).

In Bayesian linear regression, the predictive distribution of $y_* \in \mathcal{Y}$, the output value of a new input $\boldsymbol{x}_* \in \mathcal{X}$, is obtained easily as $\int p(y_*|\boldsymbol{x}_*, \boldsymbol{w}) p(\boldsymbol{w}|\{\boldsymbol{x}_i\}_{i=1}^N, \boldsymbol{y}) d\boldsymbol{w}$, which is analytically treatable. On the other hand, the GP regression requires an additional assumption to construct the predictive distribution because the marginal likelihood

(2.10) says nothing about relationships between $\boldsymbol{y}$ and y_*. In the next subsection, we see that the assumption "the regression function $f : \mathcal{X} \to \mathcal{Y}$ follows a GP" suffices to determine the predictive distribution of the GP regression.

2.2.2 Bottom-Up View of GP Modeling

To obtain the predictive distribution of the GP regression, we restart from the definition of GPs.

Definition 2.5 (Gaussian process)**.** *Let $f : \mathcal{X} \to \mathcal{Y}$ be a random function. f is said to follow a Gaussian process (GP) if the marginal distribution $p(f(\boldsymbol{x}_1), \dots, f(\boldsymbol{x}_N))$ is a multivariate Gaussian for any $N \in \mathbb{N}$ and $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N \in \mathcal{X}$.*

If f follows a GP, the marginal distribution $p(f(\boldsymbol{x}_1), \dots, f(\boldsymbol{x}_N))$ is characterized by the values of a mean function $m_X(\boldsymbol{x}) := \mathbb{E}[f(\boldsymbol{x})]$ and a covariance function $k_X(\boldsymbol{x}, \boldsymbol{x}') := \mathbb{E}[f(\boldsymbol{x})f(\boldsymbol{x}')] - \mathbb{E}[f(\boldsymbol{x})]\mathbb{E}[f(\boldsymbol{x}')]$. That is, denoting $\boldsymbol{f} = (f(\boldsymbol{x}_1), \dots, f(\boldsymbol{x}_N))^\top, \boldsymbol{m}_X = (m_X(\boldsymbol{x}_1), \dots, m_X(\boldsymbol{x}_N))^\top$, and $\boldsymbol{K}_X = (k_X(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j=1}^N$, the marginal distribution is $p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{m}_X, \boldsymbol{K}_X)$. We often denote $f \sim \mathcal{GP}(m_X, k_X)$ when f follows a GP with the mean function m_X and the convariance function k_X.

The covariance function k_X should yield a positive (semi-) definite covariance matrix $\boldsymbol{K}_X$ for any $N \in \mathbb{N}$ and $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N \in \mathcal{X}$. Since positive definite kernels satisfy this property from Definition 2.1, we can use a positive definite kernel k to define the covariance function k_X as $k = k_X$. In practice, GPs are expressive enough even without explicit modeling for the mean function m_X. We thus assume $f \sim \mathcal{GP}(0, k)$ usually.

Let us review the GP regression discussed in the previous subsection with a latent function $f \sim \mathcal{GP}(0, k)$. Consider the Bayesian generative model with a GP prior:

$$\begin{aligned} &p(y(\boldsymbol{x})|f(\boldsymbol{x}); \sigma^2) = \mathcal{N}(f(\boldsymbol{x}), \sigma^2) \text{ for any } \boldsymbol{x} \in \mathcal{X}, \\ &p(f) = \mathcal{GP}(0, k). \end{aligned}$$

If observations $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ are given, the model becomes

$$\begin{aligned} &p(y_i|f_i; \sigma^2) = \mathcal{N}(f_i, \sigma^2) \text{ for } i = 1, \dots, N, \\ &p(\boldsymbol{f}|\{\boldsymbol{x}_i\}_{i=1}^N) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{K}), \end{aligned}$$

where $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N))^\top$ and $\boldsymbol{K} = (k(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j=1}^N$. By marginalizing $\boldsymbol{f}$, we have

$$\begin{aligned} p(\boldsymbol{y}|\{\boldsymbol{x}_i\}_{i=1}^N; \sigma^2) &= \int p(\boldsymbol{y}|\boldsymbol{f}; \sigma^2) p(\boldsymbol{f}|\{\boldsymbol{x}_i\}_{i=1}^N) d\boldsymbol{f} \\ &= \mathcal{N}(\boldsymbol{0}, \boldsymbol{K} + \sigma^2 \boldsymbol{I}). \end{aligned}$$

This is exactly Equation (2.10), the marginal likelihood of the Bayesian linear regression.

We now reconsider the predictive distribution $p(y_*|\boldsymbol{x}_*, \boldsymbol{y}, \{\boldsymbol{x}_i\}_{i=1}^N; \sigma^2)$. Let $y_{N+1} = y_*, \boldsymbol{x}_{N+1} = \boldsymbol{x}_*$, and $f_{N+1} = f(\boldsymbol{x}_{N+1})$. The generative model then becomes

$$\begin{aligned} p(y_i|f_i; \sigma^2) &= \mathcal{N}(f_i, \sigma^2) \ \text{ for } i = 1, \ldots, N+1, \\ p(\boldsymbol{f}, f_{N+1}|\{\boldsymbol{x}_i\}_{i=1}^{N+1}) &= \mathcal{N}\left(\left(\begin{matrix} \boldsymbol{f} \\ f_{N+1} \end{matrix}\right) \middle| \boldsymbol{0}, \begin{pmatrix} \boldsymbol{K} & \boldsymbol{k}_*^\top \\ \boldsymbol{k}_* & k_{**} \end{pmatrix}\right), \end{aligned}$$

where $\boldsymbol{k}_* = (k(\boldsymbol{x}_1, \boldsymbol{x}_{N+1}), k(\boldsymbol{x}_2, \boldsymbol{x}_{N+1}), \ldots, k(\boldsymbol{x}_N, \boldsymbol{x}_{N+1}))^\top$ and $k_{**} = k(\boldsymbol{x}_{N+1}, \boldsymbol{x}_{N+1})$. The predictive distribution is derived as

$$\begin{aligned} p(y_{N+1}|\boldsymbol{y}, \{\boldsymbol{x}_i\}_{i=1}^{N+1}; \sigma^2) &= \frac{p(\boldsymbol{y}, y_{N+1}|\{\boldsymbol{x}_i\}_{i=1}^{N+1}; \sigma^2)}{p(\boldsymbol{y}|\{\boldsymbol{x}_i\}_{i=1}^N; \sigma^2)} \\ &= \frac{\int p(\boldsymbol{y}, y_{N+1}|\boldsymbol{f}, \boldsymbol{f}_{N+1}; \sigma^2) p(\boldsymbol{f}, f_{N+1}|\{\boldsymbol{x}_i\}_{i=1}^{N+1}) d\boldsymbol{f}}{p(\boldsymbol{y}|\{\boldsymbol{x}_i\}_{i=1}^N; \sigma^2)} \\ &= \mathcal{N}(\boldsymbol{y}^\top (\boldsymbol{K} + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{k}^*, k_{**} - \boldsymbol{k}_*^\top (\boldsymbol{K} + \sigma^2)^{-1} \boldsymbol{k}_*). \end{aligned} \tag{2.12}$$

The predictive mean of GP regression coincides with the prediction of kernel ridge regression (2.4).

2.2.3 Some Topics about GPs

Efficient Computation for GPs

Computational costs of GP methods is the key issue as in kernel methods. While the Nyström approximation is also effective for GPs, a sparse variational Gaussian process (SVGP) approach has been developed in [Titsias, 2009]. SVGP has an equivalence to the Nyström approximation Wild et al. [2021] but can take a fully Bayesian approach to obtain posterior predictive distributions.

Unsupervised Learning

The GP regression can be extended to unsupervised dimensionality reduction. Consider D-dimensional outputs $Y = (\boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(D)}) \in \mathbb{R}^{N \times D}$ and P-dimensional latent variables $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_N)^\top \in \mathbb{R}^{N \times P}$, where $P \leq D$. From a similar procedure as the GP regression, the marginal likelihood

$$p(Y|\boldsymbol{X}) = \prod_{d=1}^{D} \mathcal{N}(\boldsymbol{y}^{(d)}|\boldsymbol{0}, \boldsymbol{K} + \sigma^2 \boldsymbol{I})$$

can be obtained. The posterior of the latent variables $p(\boldsymbol{X}|Y) \propto p(Y|\boldsymbol{X})p(\boldsymbol{X})$ is then learnable with an appropriate prior $p(\boldsymbol{X})$. This method is called the Gaussian process latent variable model (GPLVM) approach [Lawrence, 2005]. We can apply a SVGP-like method for learning GPLVM [Titsias and Lawrence, 2010, Damianou et al., 2016] while a scaled conjugate gradient algorithm is used in the original GPLVM paper [Lawrence, 2005]. Wang et al. [2005, 2008] have developed a Gaussian process dynamical model (GPDM), which incorporates temporal dynamics to GPLVM. GPDM employs a GP-based vector auto-regressive prior for $p(\boldsymbol{X})$ (see Subsection 3.3.1 for details).

Connection to Kernel Methods

The equivalence of kernel ridge regression (2.4) and GP regression (2.12) implies close connections between kernel methods and GP methods. While GPs are often said to be probabilistic versions of kernel methods, the kernel methods and the GP methods have been developed separately due to their historical background; the kernel methods have been studied in the context of RKHS theory, and the GP methods have been studied via theory of stochastic processes and perspectives from probabilistic modeling. In recent years, theoretical connections between kernel methods and GPs were systematized [Kanagawa et al., 2018].

Although the connections between the kernel methods and GPs are being understood, we should be aware of differences of them sometimes. One of the phenomena specific to GPs is Driscoll's zero-one law [Driscoll, 1973], which states about the question: "do sample paths $f \sim \mathcal{GP}(0, k)$ belong to the RKHS induced by the covariance function k?" The Driscoll's zero-one law says:

- If the covariance function k is finite-dimensional (e.g., Euclidean inner product), a stochastic process $\tilde{f}$ which satisfies that $f(\boldsymbol{x}) = \tilde{f}(\boldsymbol{x})$ holds for any $\boldsymbol{x} \in \mathcal{X}$ and $\tilde{f}$ belongs to the RKHS induced by k exists, with probabilty 1.
- If k is infinite-dimensional (e.g., Gaussian kernel), f does not belong to the RKHS induced by k almost surely.

2.3 Determinantal Point Processes

A determinantal point process (DPP) is a kind of point processes and produces random subsets with diverse elements of a ground set. DPPs are alternative probabilistic models associated with positive definite kernels. We present an introduction to DPPs.

2.3.1 Point Processes

First of all, we introduce point processes without mathematical rigor. Consider a possibly uncountable ground set Ω and let $n \in \mathbb{N} \cup \{0\}$ be a random integer and $x_1, \ldots, x_n \in \Omega$ be random points. We denote the random collection as $Z = \{x_1, \ldots, x_n\}$.

Definition 2.6 (Point process[2]). *A point process η on Ω is a random measure such that*

$$\eta = \sum_{i=1}^{n} \delta_{x_i},$$

where δ_x is the Dirac measure.

For any compact $\mathcal{A} \subseteq \Omega$ we have

$$\eta(\mathcal{A}) = \sum_{i=1}^{n} \delta_{x_i}(\mathcal{A}) = |Z \cup \mathcal{A}|,$$

[2] Although such η is called a *proper* point process in [Last and Penrose, 2017], we use this as the definition of general point processes for simplicity.

which means a number of x_is that belong to $\mathcal{A}$. In other words, a point process determines which (at most countable) random point configurations on Ω tend to be realized. For example, we can consider the intensity measure defined by

$$\beta_1(\mathcal{A}) = \mathbb{E}[\eta(\mathcal{A})].$$

This represents the expected number of points, or the density, in $\mathcal{A}$.

Definition 2.6 implies that the space of all possible η is too large to write probability distributions on the space directly. Instead, we consider another way to characterize the randomness of η.

Definition 2.7 (Moment measure and joint intensity)**.** *Let $n \in \mathbb{N}$ and $\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_n$ be disjoint and compact subsets of Ω. Then, the n-th moment measure of a point process η is the measure β_m defined by*

$$\beta_n(\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n) = \mathbb{E}[\eta(\mathcal{A}_1)\eta(\mathcal{A}_2)\cdots\eta(\mathcal{A}_n)].$$

Specifically, if $\rho_n : \underbrace{\Omega \times \Omega \times \cdots \times \Omega}_{n \text{ times}} \longrightarrow \mathbb{R}$ *exists such that*

$$\beta_n(\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n) = \int_{\mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_n} \rho_n(x_1, x_2, \ldots, x_n) dx_1 dx_2 \cdots dx_n, \tag{2.13}$$

ρ_n is called an n-th joint intensity function of the point process η.

Intuitively, moment measures work like moments of a probability distribution. If we know the forms of ρ_n for all $n \in \mathbb{N}$, we may know the underlying point process.

For example, let us consider Poisson point processes, the most representative point processes, according to the above preparation.

Definition 2.8 (Poisson point process [Last and Penrose, 2017])**.** *Let η be a point process and β_1 be the intensity measure (or the first moment measure). If η satisfies the following two properties, η is said to be a Poisson point process:*

(i) For every $n \in \mathbb{N}$ and disjoint and compact subsets $\mathcal{A}_1, \ldots, \mathcal{A}_n \subseteq \Omega$, the random variables $\eta(\mathcal{A}_1), \ldots, \eta(\mathcal{A}_n)$ are independent.

(ii) For every compact $\mathcal{A} \subseteq \Omega$, $\eta(\mathcal{A}) \sim \mathrm{Poisson}(\beta_1(\mathcal{A}))$.

In addition, if $\beta_1(\mathcal{A}) = c\mathrm{Vol}(\mathcal{A}), c > 0$ for any compact $\mathcal{A} \subseteq \Omega$, η is said to be a homogeneous (or stationary) Poisson point process with the intensity c.

For a homogeneous Poisson point process we can immediately see that the n-th joint intensity is $\rho_n(x_1, \ldots, x_n) \equiv c^n$ for every $n \in \mathbb{N}$. This is because assuming $\rho_n(x_1, \ldots, x_n) \equiv c^n$ for $n = 1, 2, \ldots$, the left-hand side of (2.13) becomes

$$\begin{aligned}
\beta_n(\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n) &= \mathbb{E}[\eta(\mathcal{A}_1)\eta(\mathcal{A}_2)\cdots\eta(\mathcal{A}_n)] && \text{(from Definition 2.7)} \\
&= \mathbb{E}[\eta(\mathcal{A}_1)]\mathbb{E}[\eta(\mathcal{A}_2)]\cdots\mathbb{E}[\eta(\mathcal{A}_n)] && \text{(from (i) in Definition 2.8)} \\
&= c^n \prod_{i=1}^{n} \mathrm{Vol}(\mathcal{A}_i) && \text{(from the homogeneity)}
\end{aligned}$$

and the right-hand side is

$$\begin{aligned}
\int_{\mathcal{A}_1\times\mathcal{A}_2\times\cdots\times\mathcal{A}_n} \rho_n(x_1, x_2, \ldots, x_n) dx_1 dx_2 \cdots dx_n &= c^n \int_{\mathcal{A}_1\times\mathcal{A}_2\times\cdots\times\mathcal{A}_n} dx_1 dx_2 \cdots dx_n \\
&= c^n \prod_{i=1}^{n} \mathrm{Vol}(\mathcal{A}_i),
\end{aligned}$$

for every $n \in \mathbb{N}$ and every compact and disjoint $\mathcal{A}_1, \ldots, \mathcal{A}_n \subseteq \Omega$.

2.3.2 Determinantal Point Processes

Definition 2.9 (Determinantal point process [Hough et al., 2009])**.** *Let $k : \Omega \times \Omega \to \mathbb{R}_+$ be a positive definite kernel. A determinantal point process (DPP) with the kernel k is a point process on Ω whose joint intensities are formed as*

$$\rho_n(x_1, x_2, \ldots, x_n) = \det(\boldsymbol{K}_{[n]})$$

for any $n \in \mathbb{N}$, $x_1, x_2, \ldots, x_n \in \Omega$ and $\boldsymbol{K}_{[n]} = (k(x_i, x_j))_{i,j=1}^n$.

The following theorem gives a sufficient condition for the existence and uniqueness of DPP:

Theorem 2.10 (Soshnikov [2000], Shirai and Takahashi [2000])**.** *Let $\mathcal{K}$ be a self-adjoint integral operator determined by a kernel function k and be of locally trace class. Then, the kernel function $k(\cdot, \cdot)$ determines a DPP if and only if all the eigenvalues of $\mathcal{K}$ are in* $[0, 1]$.

If the restriction of an operator $\mathcal{K}$ to an arbitrary compact subset of Ω is of trace class, $\mathcal{K}$ is said to be locally trace class. Roughly speaking, Theorem 2.10 states that a positive definite kernel $k(\cdot, \cdot)$ defines a DPP under appropriate scaling which ensures the resulting probabilities in $[0, 1]$.

In the context of machine learning, DPPs on a finite ground set $\mathcal{Y} = \{1, 2, \dots, N\}$ are typically considered. On the finite ground set $\mathcal{Y}$, a point process $\mathbb{P}(\cdot)$ is a DPP with a kernel matrix $\boldsymbol{K} \in \mathbb{S}_+^N$ if

$$\mathbb{P}(\mathcal{S} \subseteq \mathcal{A}) = \det([\boldsymbol{K}]_{\mathcal{S}})$$

for a random subset $\mathcal{A} \subseteq \mathcal{Y}$ drawn by $\mathbb{P}$ and an arbitrary $\mathcal{S} \subseteq \mathcal{Y}$. $[\boldsymbol{K}]_{\mathcal{S}} = (K_{ij})_{i,j \in \mathcal{S}} \in \mathbb{S}_+^{|\mathcal{S}|}$ denotes the principal submatrix of $\boldsymbol{K}$ and the kernel matrix $\boldsymbol{K}$ must be $\boldsymbol{O} \preceq \boldsymbol{K} \preceq \boldsymbol{I}$ from an analogy with the DPPs on a general ground set[3].

For instance, the inclusion probability of $i \in \mathcal{Y}$ is

$$\mathbb{P}(\{i\} \subseteq \mathcal{A}) = \det(K_{ii}) = K_{ii} = k(i, i).$$

For $|\mathcal{S}| = 2$, we have

$$\begin{aligned}\mathbb{P}(\{i, j\} \subseteq \mathcal{A}) = \det \begin{pmatrix} K_{ii} & K_{ij} \\ K_{ij} & K_{jj} \end{pmatrix} &= K_{ii}K_{jj} - K_{ij}^2 \\ &= \mathbb{P}(\{i\} \subseteq \mathcal{A})\mathbb{P}(\{j\} \subseteq \mathcal{A}) - K_{ij}^2.\end{aligned} \tag{2.14}$$

A DPP on a finite ground set has an alternative representation called the $\boldsymbol{L}$-ensemble [Borodin and Rains, 2005], which defines the occurrence probability of a random subset

[3] We use $\prec, \preceq, \succ$, and $\succeq$ in the sense of positive (semi-)definite ordering.

$\mathcal{A} \subseteq \mathcal{Y}$ as

$$P_{\boldsymbol{L}}(\mathcal{A}) = \frac{\det([\boldsymbol{L}]_{\mathcal{A}})}{\det(\boldsymbol{L} + \boldsymbol{I})}, \tag{2.15}$$

where $\boldsymbol{L} \in \mathbb{S}_{+}^{N}$ is a positive semidefinite kernel matrix. We can commute between $\boldsymbol{K}$ and $\boldsymbol{L}$ using the equation $\boldsymbol{K} = \boldsymbol{L}(\boldsymbol{L} + \boldsymbol{I})^{-1}$ or its inversion $\boldsymbol{L} = \boldsymbol{K}(\boldsymbol{I} - \boldsymbol{K})^{-1}$ if $\boldsymbol{I} - \boldsymbol{K}$ is invertible.

2.3.3 Properties of DPPs

Negative Dependence

One of the most notable properties of DPPs is negative dependence, which encourages inter-element repulsion within the random subsets. Borcea et al. [2009] introduced strongly Rayleigh measures, which include DPPs, as a class of measures that suffice for negative dependence. The Ph.D. thesis by [Mariet, 2019] provides a good review about negative dependence.

Some characterizations exists for negative dependence. When

$$\mathbb{P}(\{i\} \subseteq \mathcal{A})\mathbb{P}(\{j\} \subseteq \mathcal{A}) \geq \mathbb{P}(\{i, j\} \subseteq \mathcal{A}) \text{ for all } i, j \in \mathcal{Y}$$

holds, one is said to satisfy the pairwise negative correlation. The negative lattice condition

$$P(\mathcal{A}_1)P(\mathcal{A}_2) \geq P(\mathcal{A}_1 \cup \mathcal{A}_2)P(\mathcal{A}_1 \cap \mathcal{A}_2) \text{ for all } \mathcal{A}_1, \mathcal{A}_2 \subseteq \mathcal{Y}$$

is also defined other than the pairwise negative correlation. The negative lattice condition is equivalent to the log-submodularity of the distribution over $2^{\mathcal{Y}}$.

Because DPPs satisfy both the pairwise negative correlation and negative lattice condition, the sampled subsets from a DPP tend to contain diverse items. In fatc, Equation (2.14) represents the negative dependence of DPPs with the inter-element repulsion $K_{ij}^2 \geq 0$.

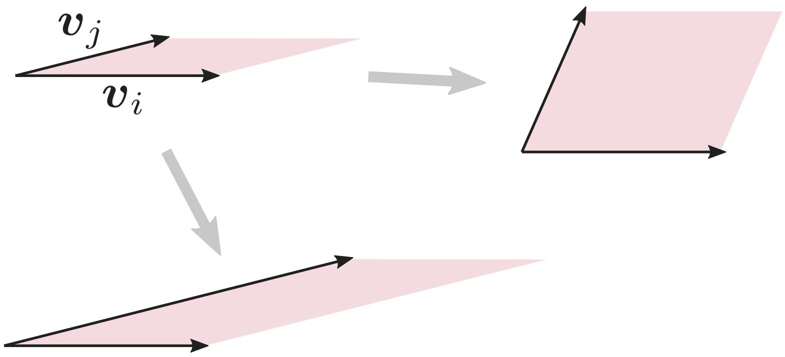


Figure 2.1: Geometry of DPPs. Larger norms and larger angles of feature vectors increase the occurrence probability.

Geometric Interpretation

We can see the negative dependence of DPPs from a geometric view. Now, we consider the $\boldsymbol{L}$-ensemble (2.15). Without loss of generality, we write the kernel matrix as $\boldsymbol{L} = \boldsymbol{V}\boldsymbol{V}^\top$ with some $\boldsymbol{V} \in \mathbb{R}^{N \times D}, D \in \mathbb{N}$.

Denoting the i-th row vector of $\boldsymbol{V}$ as $\boldsymbol{v}_i$, we can regard $\boldsymbol{v}_i$ as the feature vector of $i \in \mathcal{Y}$ and $L_{ij} = \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle$ means the similarity of i-th and j-th feature vectors. The occurrence probability of $\mathcal{A} \subseteq \mathcal{Y}$ is

$$P_{\boldsymbol{L}}(\mathcal{A}) \propto \det([\boldsymbol{L}]_{\mathcal{A}}) = \mathrm{Vol}^2(\{\boldsymbol{v}_i\}_{i \in \mathcal{A}}),$$

where $\mathrm{Vol}(\{\boldsymbol{v}_i\}_{i \in \mathcal{A}})$ is the $|\mathcal{A}|$-dimensional volume of the parallelepiped spanned by the feature vectors $\{\boldsymbol{v}_i\}_{i \in \mathcal{A}}$. Figure 2.1 illustrates the geometry of the $\boldsymbol{L}$-ensemble in the case $|\mathcal{A}| = 2$. The realization $\mathcal{A} = \{i, j\}$ is more likely to appear as the norms $\|\boldsymbol{v}_i\|$ and $\|\boldsymbol{v}_j\|$ become larger. Additionally, we can find that the closer $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are to orthogonal, the larger $P(\{i, j\})$ is induced. This is considered to be the mechanism that results in the negative dependence of DPPs.

Other Properties

Notably DPPs support analytic expressions about some elementary probabilistic operations. For example, suppose that $\mathcal{A}_1 \subseteq \mathcal{Y}$ and $\mathcal{A}_2 \subseteq \mathcal{Y}$ are disjoint, and let $\mathcal{A}_1$ be observed. Then, the conditional probability of $\mathcal{A}_1 \cup \mathcal{A}_2$ is $P_{\boldsymbol{L}}(\mathcal{A}_1 \cup \mathcal{A}_2 | \mathcal{A}_1) \propto \det([\boldsymbol{L}]_{\mathcal{A}_1 \cup \mathcal{A}_2})$, and this is also the DPP [Borodin and Rains, 2005]. Marginal probabilities can also be computed with K as done in (2.14). See [Kulesza and Taskar, 2012] for the other operations.

Sampling from a DPP on a possibly infinite ground set can be done by a simple algorithm [Hough et al., 2006, Theorem 7]. MCMC samplers have also been developed [Anari et al., 2016, Li et al., 2016, Derezinski et al., 2019] while a neural network-based fast approximate sampler for DPPs is proposed [Mariet et al., 2019b]. We also note that `DPPy`, the sampling toolbox for DPPs in Python, is developed by [Gautier et al., 2019].

3
Gaussian Process Koopman Mode Decomposition

3.1 Background

Many real-world phenomena are observed as multivariate (time) series data. Although they sometimes appear to be disorderly, the obtained data may be governed by some intrinsic law. Because such laws are expressed as dynamical systems in many fields, the development of data-driven approaches to understand unknown dynamical systems is probably inevitable.

One data-driven strategy for dynamical systems is to employ state space models, classically represented by the Kalman filter [Kalman, 1960], ensemble Kalman filter [Evensen, 2003], particle filter [Gordon et al., 1993, Kitagawa, 1996], and 4D-Var [Lewis and Derber, 1985, Dimet and Talagrand, 1986]. An alternative approach is mode decomposition, which extracts some oscillating components from data. If some background knowledge validates the assumption of a dynamical system, we can comprehend the data by estimating time-invariant parameters, including the modes.

Koopman mode decomposition (KMD) enables us to specify the quantities to be estimated on the basis of the Koopman operator theory [Mezić, 2005, Rowley et al., 2009]. Although only limited special systems enable analytic calculations of the quantities, dynamic mode decomposition (DMD) provides a general data-driven algorithm to approximate them [Rowley et al., 2009, Schmid, 2010]. DMD is primitively divided into two types: the Arnoldi type [Rowley et al., 2009] and SVD-based type [Schmid, 2010]. Both types give a simple linear approximation of the dynamics on an observation space, thus various DMD extensions have been developed in the last decade [Jovanović et al., 2014, Dawson et al., 2016, Le Clainche and Vega, 2017, Héas and Herzet, 2020]. To overcome the limitations of linear approximations, some nonlinear extensions of DMD have been proposed on the basis of user-defined bases [Williams et al., 2015a], kernel methods [Williams et al., 2015b, Kawahara, 2016], or neural networks [Takeishi et al., 2017a]. Nonetheless, nonlinear probabilistic generative models of KMD have not yet been studied, as mentioned in Section 3.1.2.

3.1.1 Contributions

In this chapter, we develop a nonlinear generative model for KMD with an unsupervised Gaussian process (GP) named Gaussian process Koopman mode decomposition (GPKMD). An existing unsupervised GP method for dynamical systems known as Gaussian process dynamical model (GPDM) [Wang et al., 2005] already exists. The GPDM was derived from the Gaussian process latent variable model (GPLVM) [Lawrence, 2005], which is the GP form of probabilistic principal component analysis (probabilistic PCA), and can be viewed as a GP-based extension of an autoregressive model. Whereas GPLVM and GPDM only focus on dimensionality reduction or learning nonlinear mappings from a latent space to an observation space, our method can be used to estimate the latent variables and quantities of KMD simultaneously.

This work has the following main contributions:

- We provide a novel perspective of KMD through the GP-based nonlinear generative model named GPKMD. The generative modeling enables us to estimate not only the quantities specified by KMD but also the latent variables and enables us to obtain richer information from estimands.

- We propose an efficient computing strategy for GPKMD using low-rank approximations of Gram matrices and matrix diagonalization. We show that the complexity of our strategy is markedly superior to the existing one.

- We demonstrate our proposed method on synthetic data generated from a nonlinear limit cycle and a real-world epidemiological dataset. We show the usefulness of the proposed method for interpreting the data from various viewpoints.

3.1.2 Related Works

3.1.3 Gaussian Processes and KMD

In previous works, researchers have attempted to connect KMD and GP regressions. Masuda et al. [2019] proposed a GP-based algorithm for Arnoldi-type DMD. This algorithm determines the coefficients of the companion matrix based on the prediction by GP regression, which is conditioned by past observations. Although the method employs GP regression, it requires a posteriori deterministic matrix factorization processes to obtain the Koopman eigenvalues and modes. Therefore, the advantages of probabilistic methods and interpretability are limited. Lian and Jones [2020] studied a model predictive control method based on Koopman operator theory. Because the work focused on control design, they did not discuss an inference framework for Koopman quantities.

Estimating Koopman quantities can be regarded as an inverse problem. This perspective implies that KMD is essentially an unsupervised task; therefore, as a complementary to the existing works, we take an unsupervised approach.

3.1.4 Bayesian Models of KMD

In some Bayesian models, KMD (or DMD) is treated as unsupervised learning. Takeishi et al. [2017b] proposed Bayesian DMD and an efficient sampling algorithm for the posterior. In the Bayesian DMD model, each output of the Koopman eigenfunction is parameterized as a scalar-valued i.i.d. random variable as seen in Subsection 3.2.3. However, this simplification may discard important structures in the eigenfunctions and latent variables. To alleviate this shortcoming, the Bayesian DMD with variational matrix factorization (BDMD-VMF) model was developed [Kawashima et al., 2021], in which an explicit treatment of the eigenfunctions is avoided. Moreover, BDMD-VMF employs VMF [Lim and Teh, 2007, Nakajima and Sugiyama, 2011] to determine its prior and marginalize its higher-dimensional parameters; thus, the computational stability is improved even for incomplete observations.

In this chapter, we develop a GP-based generative model of KMD as an extension of Bayesian DMD. Whereas both Bayesian DMD and BDMD-VMF are based on linear parameterizations of the output of the Koopman eigenfunctions, not the latent variables $\{\boldsymbol{x}_t\}$, our model explicitly incorporates the latent variables as model parameters (i.e., random variables). To the best of our knowledge, this is the first work enabling the latent variables to be directly estimated from observations in the framework of KMD.

3.2 Koopman Mode Decomposition and Computational Methods

We give a brief introduction of existing methods in relation to our proposal.

3.2.1 Koopman Mode Decomposition

Koopman mode decomposition (KMD) is a framework to transform multidimensional series data into a tractable sum-of-modes representation. We provide a brief introduction to KMD.

Let the latent variables $\boldsymbol{x}_t \in \mathcal{X} \subseteq \mathbb{R}^P$ be evolved deterministically by an unknown map $\boldsymbol{f} : \mathcal{X} \to \mathcal{X}$,

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t). \tag{3.1}$$

Observations that we can treat are obtained through an observable $\mathcal{G} \ni g : \mathcal{X} \to \mathbb{C}$ as $g(\boldsymbol{x}_t)$, where $\mathcal{G}$ is an appropriate complex-valued function space. The Koopman operator $\mathcal{K} : \mathcal{G} \to \mathcal{G}$ is defined as an operator that maps the observable at t to that at $t + 1$:

$$(\mathcal{K}g)(\boldsymbol{x}_t) = (g \circ \boldsymbol{f})(\boldsymbol{x}_t) = g(\boldsymbol{x}_{t+1}).$$

Although we considered the latent dynamics $\boldsymbol{f}$ above, the Koopman operator $\mathcal{K}$ can also describe the evolution of a system on the function space $\mathcal{G}$. Despite the nonlinearity of f, the Koopman operator is linear owing to its lifting to the infinite-dimensional space. This property permits the spectral decomposition of $\mathcal{K}$,

$$\mathcal{K}\phi_k = \lambda_k \phi_k, \tag{3.2}$$

where $\lambda_k \in \mathbb{C}$ and $\phi_k : \mathcal{X} \to \mathbb{C}$ are the k-th Koopman eigenvalue and the corresponding Koopman eigenfunction, respectively. Suppose that there are D distinct observables $g_1, \dots, g_D$ such that $g_d \in \mathcal{G}, d = 1, \dots, D$, then we define a D-dimensional observation $\boldsymbol{y}_t = \boldsymbol{g}(\boldsymbol{x}_t) = (g_1(\boldsymbol{x}_t), \dots, g_D(\boldsymbol{x}_t))^\top \in \mathbb{C}^D$. Assuming that the D-dimensional observable $\boldsymbol{g}$ is expanded by Koopman eigenfunctions $\{\phi_k\}$, we obtain

$$\boldsymbol{y}_t = \boldsymbol{g}(\boldsymbol{x}_t) = \sum_{k=1}^{\infty} \phi(\boldsymbol{x}_t)\boldsymbol{w}_k, \tag{3.3}$$

where $\boldsymbol{w}_k \in \mathbb{C}^D$ is the k-th coefficient called the Koopman mode. By applying spectral decomposition (3.2) to (3.3), observations can be transformed recurrently as

$$\begin{aligned}\boldsymbol{y}_t = \boldsymbol{g}(\boldsymbol{x}_t) = (\mathcal{K}\boldsymbol{g})(\boldsymbol{x}_{t-1}) &= \sum_{k=1}^{\infty} (\mathcal{K}\phi_k)(\boldsymbol{x}_{t-1})\boldsymbol{w}_k \\ &= \sum_{k=1}^{\infty} \lambda_k \phi_k(\boldsymbol{x}_{t-1})\boldsymbol{w}_k \\ &= \cdots = \sum_{k=1}^{\infty} \lambda_k^t \phi_k(\boldsymbol{x}_0)\boldsymbol{w}_k. \end{aligned} \tag{3.4}$$

Note that the Koopman operator $\mathcal{K}$ has not only the discrete spectra but also continuous spectra because of the infinite dimensionality. Roughly speaking, discrete and continuous spectra represent the quasi-periodic and chaotic parts of the evolving process, respectively [Mezić, 2005, Colbrook et al., 2023, Colbrook and Townsend, 2024]. Given observations $(\boldsymbol{y}_0,)\boldsymbol{y}_1, \dots, \boldsymbol{y}_T$, we can unravel the hidden quasi-periodic dynamics governing the system by estimating the Koopman quantities $\{\lambda_k\}$, $\{\phi_k\}$, and $\{\boldsymbol{w}_k\}$ in (3.4), instead of f. KMD is the framework used to understand the data-generating system with this scheme. The inferable quantities depend on the algorithm; for example, DMD approximates f by low-rank linear dynamics and provides the finite sets $\{\lambda_k\}$ and $\{\boldsymbol{w}_k\}$.

3.2.2 Dynamic Mode Decomposition

Dynamic mode decomposition (DMD), proposed in [Rowley et al., 2009, Schmid, 2010], is the representative method to obtain Koopman quantities from a data series. By defining $b_k := \phi_k(\boldsymbol{x}_0)$ and truncating the infinite sum up to K, Equation (3.4) becomes

$$\boldsymbol{y}_t \approx \sum_{k=1}^{K} \lambda_k^t \boldsymbol{w}_k b_k = \boldsymbol{W}\boldsymbol{\Lambda}^t \boldsymbol{b}$$

for $t = 0, \dots, T$, where $\boldsymbol{W} := (\boldsymbol{w}_1, \dots, \boldsymbol{w}_K)$, $\boldsymbol{\Lambda} := \mathrm{diag}(\lambda_1, \dots, \lambda_K)$, and $\boldsymbol{b} := (b_1, \dots, b_K)^\top$. We then define

$$\boldsymbol{A} := \boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{W}^+, \tag{3.5}$$

where $\boldsymbol{W}^+$ indicates the pseudo-inverse of $\boldsymbol{W}$. When $K \leq D$ and $\boldsymbol{W} \in \mathbb{C}^{D\times K}$ is full-rank: $\mathrm{rank}(\boldsymbol{W}) = K$, we have $\boldsymbol{A}^t = \boldsymbol{W}\boldsymbol{\Lambda}^t\boldsymbol{W}^+$ and $\boldsymbol{A}^t\boldsymbol{W} = \boldsymbol{W}\boldsymbol{\Lambda}^t$. Now, recalling $\boldsymbol{W}\boldsymbol{b} \approx \boldsymbol{y}_0$, we can confirm that

$$\boldsymbol{y}_t \approx \boldsymbol{W}\boldsymbol{\Lambda}^t\boldsymbol{b} = \boldsymbol{A}^t\boldsymbol{W}\boldsymbol{b} \approx \boldsymbol{A}^t\boldsymbol{y}_0 \tag{3.6}$$

holds for $t = 0, \dots T$. Equation (3.6) implies that the solution of

$$\min_{A\in C^{D\times D}} \sum_{t=1}^{T} \|\boldsymbol{y}_t - \boldsymbol{A}\boldsymbol{y}_{t-1}\|^2 \tag{3.7}$$

has entire information about the (approximated) system. Indeed, we can see that the eigenvalues and eigenvectors of $\boldsymbol{A}$ can be regarded as estimated Koopman eigenvalues $\{\lambda_k\}_{k=1}^K$ and Koopman modes $\{\boldsymbol{w}_k\}_{k=1}^K$, respectively, from (3.5). The DMD algorithm provided in [Schmid, 2010] produces an efficient numerical method to find the dominant K eigenvalues and corresponding eigenvectors of the solution matrix of (3.7).

3.2.3 Bayesian Dynamic Mode Decomposition

Takeishi et al. [2017b] proposed a Bayesian variant of DMD. Let (Y_t, Y'_{t-1}) be a pair of $\mathbb{C}^D$-valued random vectors for $t = 1, \dots, T$. Given observations $\boldsymbol{y}_1, \dots, \boldsymbol{y}_T \in \mathbb{C}^D$, Bayesian DMD models its likelihood as:

$$
\begin{aligned}
&p(\boldsymbol{y}_t|\{\lambda_k\}_k, \{\phi_{k,t}\}_{k,t}, \{\boldsymbol{w}_k\}_k, \sigma^2) \\
&\quad := \frac{1}{Z} p_{Y_t, Y'_{t-1}}(\boldsymbol{y}_t, \boldsymbol{y}'_t|\{\lambda_k\}_k, \{\phi_{k,t}\}_{k,t}, \{\boldsymbol{w}_k\}_k, \sigma^2)\mathbb{1}(\boldsymbol{y}_t = \boldsymbol{y}'_t),
\end{aligned}
$$

where Z is an appropriate normalizing constant and

$$
\begin{aligned}
p_{Y_t}(\boldsymbol{y}_t|\{\phi_{k,t}\}_{k,t}, \{\boldsymbol{w}_k\}_k, \sigma^2) &= \mathcal{CN}\left(\boldsymbol{y}_t \left| \sum_{k=1}^{K} \phi_{k,t}\boldsymbol{w}_k, 2\sigma^2\boldsymbol{I}\right.\right), \\
p_{Y'_{t-1}}(\boldsymbol{y}_t|\{\lambda_k\}_k, \{\phi_{k,t}\}_{k,t}, \{\boldsymbol{w}_k\}_k, \sigma^2) &= \mathcal{CN}\left(\boldsymbol{y}_t \left| \sum_{k=1}^{K} \lambda_k\phi_{k,t-1}\boldsymbol{w}_k, 2\sigma^2\boldsymbol{I}\right.\right),
\end{aligned} \tag{3.8}
$$

are conditionally independent densities for $t = 1, \ldots, T$[1]. Note that $\mathcal{CN}(\cdot, \cdot)$ denotes a complex normal distribution. Takeishi et al. [2017b] developed an efficient Gibbs sampler for the Bayesian DMD under appropriate prior distributions.

What does the likelihood (3.8) mean? Now, we can find that

$$
\begin{aligned}
&p(\boldsymbol{y}_t|\{\lambda_k\}_k, \{\phi_{k,t}\}_{k,t}, \{\boldsymbol{w}_k\}_k, \sigma^2) \\
&\quad = \mathcal{CN}\left(\boldsymbol{y}_t \left| \sum_{k=1}^{K} \phi_{k,t}\boldsymbol{w}_k, 2\sigma^2\boldsymbol{I}\right.\right) \mathcal{CN}\left(\boldsymbol{y}_t \left| \sum_{k=1}^{K} \lambda_k\phi_{k,t-1}\boldsymbol{w}_k, 2\sigma^2\boldsymbol{I}\right.\right), \\
&\quad \propto \mathcal{CN}\left(\boldsymbol{y}_t \left| \frac{1}{2}\left(\sum_{k=1}^{K} \phi_{k,t}\boldsymbol{w}_k + \sum_{k=1}^{K} \lambda_k\phi_{k,t-1}\boldsymbol{w}_k\right), \sigma^2\boldsymbol{I}\right.\right),
\end{aligned} \tag{3.9}
$$

where we used the following relation:

$$
\mathcal{CN}(\boldsymbol{x}|\boldsymbol{\mu}_1, \sigma_1^2\boldsymbol{I})\mathcal{CN}(\boldsymbol{x}|\boldsymbol{\mu}_2, \sigma_2^2\boldsymbol{I}) \propto \mathcal{CN}\left(\boldsymbol{x} \left| \frac{\sigma_2^2\boldsymbol{\mu}_1 + \sigma_1^2\boldsymbol{\mu}_2}{\sigma_1^2 + \sigma_2^2}, \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right.\right).
$$

[1] The scaling of the variances are different with the original paper, but this does not lose generality.

Equation (3.9) implies that the expectation of the likelihood (3.8) is

$$\mathbb{E}[\boldsymbol{y}_t | \{\lambda_k\}_k, \{\phi_{k,t}\}_{k,t}, \{\mathbf{w}_k\}_k, \sigma^2] = \frac{1}{2}\left(\sum_{k=1}^{K} \phi_{k,t}\mathbf{w}_k + \sum_{k=1}^{K} \lambda_k \phi_{k,t-1}\mathbf{w}_k\right),$$

and it encourages to be

$$\boldsymbol{y}_t \approx \sum_{k=1}^{K} \phi_{k,t}\mathbf{w}_k \approx \sum_{k=1}^{K} \lambda_k \phi_{k,t-1}\mathbf{w}_k, \quad \text{for } t = 1, \ldots, T. \tag{3.10}$$

The approximation (3.10) has the similar form with the KMD (3.4). We can see that Bayesian DMD truncates the infinite sums in KMD by finite ones as in DMD, and each value of the eigenfunction $\phi_k(\boldsymbol{x}_t)$ is treated as a random variable $\phi_{k,t}$. While $\{\phi_{k,t}\}_{k,t}$ are i.i.d. random variables in Bayesian DMD, some structural assumption is considered to be more suitable rather than i.i.d. modeling. That is, $\phi_{k,t}$ and $\phi_{k,t+1}$ may take close values if the underlying eigenfunction $\phi_k(\cdot)$ is not ill-shaped. We introduce Gaussian processes to overcome this issue in the next section.

3.3 Gaussian Process Koopman Mode Decomposition

Gaussian processes (GPs) are representative nonparametric methods for learning nonlinear mappings from an input space $\mathcal{X}$ to an output space $\mathcal{Y}$. By formulating KMD as an unsupervised GP, we establish a nonlinear generative model of KMD.

Let $Y = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T) \in \mathbb{C}^{D\times T}$ be the data matrix and $\boldsymbol{X} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_T) \in \mathbb{R}^{P\times(T+1)}$ be the latent variables. We start by assuming that the value of each Koopman eigenfunction ϕ_k evaluated as any $\boldsymbol{x} \in \mathbb{R}^P$ is represented as the inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ on a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$. We then expand as

$$\phi_k(\boldsymbol{x}) = \langle \boldsymbol{b}_k, \boldsymbol{\psi}(\boldsymbol{x})\rangle_{\mathcal{H}} = \sum_l b_{kl}\psi_l(\boldsymbol{x})$$

using coefficients $\boldsymbol{b}_k = (b_{k1}, b_{k2}, \ldots) \in \mathcal{H}$ and the feature map $\boldsymbol{\psi} = (\psi_1, \psi_2, \ldots) : \mathcal{X} \to \mathcal{H}$. We define the likelihood of KMD by incorporating the equalities (3.3) and (3.4) up to the first-order,

$$p(\boldsymbol{y}_t|\{\boldsymbol{x}_t\},\{\lambda_k\},\{\boldsymbol{w}_k\},\{b_{kl}\},\sigma^2) := \frac{1}{Z}\mathcal{CN}\left(\boldsymbol{y}_t\middle|\sum_{k=1}^{K}\left(\sum_l b_{kl}\psi_l(\boldsymbol{x}_t)\right)\boldsymbol{w}_k,\sigma^2\boldsymbol{I}\right)$$
$$\times\,\mathcal{CN}\left(\boldsymbol{y}_t\middle|\sum_{k=1}^{K}\lambda_k\left(\sum_l b_{kl}\psi_l(\boldsymbol{x}_{t-1})\right)\boldsymbol{w}_k,\sigma^2\boldsymbol{I}\right)\quad(3.11)$$

which coincides with Bayesian DMD (3.9), and Z denotes a normalizing constant. In (3.11), the countably infinite summations of the modes are truncated at K. The expansion coefficients $\{b_{kl}\}$ can be marginalized out from each $\mathcal{CN}(\cdot,\cdot)$ in the likelihood (3.11): with the i.i.d. prior $p(b_{kl}) = \mathcal{CN}(b_{kl}|0,\sigma_b^2/2) \propto \mathcal{CN}(b_{kl}|0,\sigma_b^2)^2$. We then obtain the following marginalized likelihood (see Section 3.7 for derivation details):

$$p(Y|X,\boldsymbol{\Lambda},\boldsymbol{W},\sigma^2,\sigma_b^2) \propto \mathcal{CN}(\mathrm{vec}(Y)|\boldsymbol{0},\sigma^2\boldsymbol{I}+\sigma_b^2(\boldsymbol{K}_1\otimes\boldsymbol{W}\boldsymbol{W}^*))$$
$$\times\,\mathcal{CN}(\mathrm{vec}(Y)|\boldsymbol{0},\sigma^2\boldsymbol{I}+\sigma_b^2(\boldsymbol{K}_0\otimes\boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{\Lambda}^*\boldsymbol{W}^*)),\qquad(3.12)$$

where $\boldsymbol{W} = (\boldsymbol{w}_1,\ldots,\boldsymbol{w}_K)$ and $\boldsymbol{\Lambda} = \mathrm{diag}(\{\lambda_k\}_{k=1}^K)$. $\boldsymbol{K}_1$ and $\boldsymbol{K}_0$ are $T\times T$ Gram matrices consisting of $\{\boldsymbol{x}_t\}_{t=1}^T$ and $\{\boldsymbol{x}_t\}_{t=0}^{T-1}$ with a positive definite kernel $k(\boldsymbol{x},\boldsymbol{x}') = \langle\boldsymbol{\psi}(\boldsymbol{x}),\boldsymbol{\psi}(\boldsymbol{x}')\rangle_{\mathcal{H}}$, respectively. $\boldsymbol{W}^*$ denotes the Hermitian transpose of $\boldsymbol{W}$. The marginalized likelihood (3.12) appears unnatural because it is divided into two terms, but we can merge them into a single zero-mean $\mathcal{CN}(\cdot,\cdot)$. Since the covariance matrices of the joint likelihood (3.12) are formed by the Gram matrices of latent variables, we obtain the GP formulation for KMD. We define (3.12) as the likelihood of our proposal, Gaussian process Koopman mode decomposition (GPKMD).

3.3.1 Prior Distributions for GPKMD

We should also consider rational priors for the parameters $X,\boldsymbol{W},\boldsymbol{\Lambda},\sigma^2$, and σ_b^2. Similar to the configuration of KMD (3.1), GPKMD should incorporate latent dynamics explicitly as its prior in a probabilistic sense. Thus, we adopt a GPDM-inspired prior for the latent variable $\boldsymbol{X}$ [Wang et al., 2005]. That is, denoting $\boldsymbol{X}_1 = (\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T) \in \mathbb{R}^{P\times T}$ and a Gram matrix consisting of $\{\boldsymbol{x}_t\}_{t=0}^{T-1}$ with a kernel function $k_x(\cdot,\cdot)$ by $\boldsymbol{K}_X$, we use

$$p(X) = \mathcal{N}(\boldsymbol{x}_0|\boldsymbol{0}, s_x^2\boldsymbol{I})\mathcal{MN}(X_1|\boldsymbol{O}, \boldsymbol{I}, \boldsymbol{K}_X + s_x^2\boldsymbol{I}) \tag{3.13}$$

for the prior $p(X)$. Here, $\mathcal{MN}(\cdot,\cdot,\cdot)$ denotes a matrix normal distribution. Note that the prior can be regarded as a GP extension of the first-order autoregressive model. Unless there is a particular reason, it is reasonable to employ simple priors for other parameters, such as

$$\begin{aligned} p(w_{dk}) &= \mathcal{CN}(w_{dk}|0, s_w^2), \\ p(\lambda_k) &= \mathcal{CN}(\lambda_k|0, s_\lambda^2), \\ p(\sigma^2) &= \mathrm{InvGamma}(\sigma^2|\alpha,\beta), \\ p(\sigma_b^2) &= \mathrm{InvGamma}(\sigma_b^2|\alpha_b,\beta_b). \end{aligned}$$

3.4 Scalable Inference

In theory, the posterior or its point estimates of the GPKMD parameters can be obtained using the marginal likelihood (3.12) with appropriate priors. However, the GPKMD likelihood contains very large $DT \times DT$-sized covariance matrices, which inhibit scalable inference. Straightforward computations of the GPKMD likelihood (3.12) and its gradients require an extremely high computational cost of $O(D^3T^3)$. Hereafter, we tackle the scalability of GPKMD. We only consider the first $\mathcal{CN}(\cdot,\cdot)$ in (3.12) for simplicity in this section, but the same approach applies to the second $\mathcal{CN}(\cdot,\cdot)$.

3.4.1 Stegle's Method

Multioutput or multitask GPs often have Kronecker-structured covariance matrices, and are sometimes called Kronecker GPs [Stegle et al., 2011]. GPKMD can be considered a type of Kronecker GP. Stegle et al. [2011] and Rakitsch et al. [2013] proposed an efficient inference method for Kronecker GPs using an eigendecomposition-based trick. First, consider the eigendecomposition $\boldsymbol{K}_1 = \boldsymbol{U}_K\boldsymbol{S}_K\boldsymbol{U}_K^\top$, $\boldsymbol{W}\boldsymbol{W}^* = \boldsymbol{U}_W\boldsymbol{S}_W\boldsymbol{U}_W^*$. Following Stegle's method, the inversion of the GPKMD covariance matrix is exactly transformed into

$$
\begin{aligned}
&(\sigma^2 \boldsymbol{I} + \sigma_b^2(\boldsymbol{K}_1 \otimes \boldsymbol{W}\boldsymbol{W}^*))^{-1} \\
&= \underbrace{(\boldsymbol{U}_K \otimes \boldsymbol{U}_W)}_{DT \times DT} \underbrace{(\sigma^2 \boldsymbol{I} + \sigma_b^2(\boldsymbol{S}_K \otimes \boldsymbol{S}_W))^{-1}}_{DT \times DT \text{ (diagonal)}} \underbrace{(\boldsymbol{U}_K \otimes \boldsymbol{U}_W)^*}_{DT \times DT}.
\end{aligned}
$$

Because the matrix to be inverted is reformed into a diagonal matrix, the complexity of the inversion is reduced to $O(D^3 + T^3)$, which is dominated by the eigendecomposition for $\boldsymbol{K}_1$ and $\boldsymbol{W}\boldsymbol{W}^*$. $\log\det(\cdot)$ is similarly computed as

$$
\log\det(\sigma^2 \boldsymbol{I} + \sigma_b^2(\boldsymbol{K}_1 \otimes \boldsymbol{W}\boldsymbol{W}^*)) = \log\det\underbrace{(\sigma^2 \boldsymbol{I} + \sigma_b^2(\boldsymbol{S}_K \otimes \boldsymbol{S}_W))}_{DT \times DT \text{ (diagonal)}},
$$

and the gradients of the likelihood can also be converted to reduced forms.

Stegle's method is effective for GPKMD; however, we still have some considerations:

- For the interpretability, we often use a small number of Koopman modes, K, typically about 5–30. For $K \ll D$, the diagonal elements of the eigenvalue matrix $\boldsymbol{S}_W$ are sparse since $\operatorname{rank}(\boldsymbol{W}\boldsymbol{W}^*) = K$. This implies the possibility of further reducing in the computational cost.

- The space complexity of Stegle's method is $O(D^2 + T^2)$. For large D or/and T (e.g., $> 100,000$), ordinary computers may run out of memory.

3.4.2 Low-rank Approximations

In kernel methods, Gram matrices can be well approximated by low-rank matrices in many practical cases. Bonilla et al. [2007] proposed an efficient prediction strategy for multitask GPs by applying low-rank approximations to Gram matrices. We propose a considerably more efficient strategy for various computations of GPKMD by combining the above-explained Stegle's method and low-rank approximations.

By applying an appropriate algorithm (e.g., incomplete Cholesky decomposition or the Nyström method [Drineas and Mahoney, 2005]), we can approximate the Gram matrix as $\boldsymbol{K}_1 \approx \boldsymbol{R}\boldsymbol{R}^\top$, where $\boldsymbol{R} \in \mathbb{R}^{T \times S}$ for $S < T$. If the Nyström method (described in Subsection 2.1.3) is employed, we can obtain $\boldsymbol{C} \in \mathbb{R}^{T \times S}$ and $\Omega \in \mathbb{R}^{S \times S}$

such that $\boldsymbol{K}_1 \approx \boldsymbol{C}\Omega\boldsymbol{C}^\top$. Then, the eigendecomposition of the symmetric matrix Ω enables us to obtain the desired matrix $\boldsymbol{R}$. Subsequently, by using the thin SVD $\boldsymbol{R} = \boldsymbol{U}_K\Sigma_K\boldsymbol{V}_K^T$, $\boldsymbol{W} = \boldsymbol{U}_W\Sigma_W\boldsymbol{V}_W^*$ and the Woodbury identity, the inverse covariance matrix of GPKMD is approximated as

$$\begin{aligned}(\sigma^2\boldsymbol{I} + \sigma_b^2(\boldsymbol{K}_1 \otimes \boldsymbol{W}\boldsymbol{W}^*))^{-1} &\approx (\sigma^2\boldsymbol{I} + \sigma_b^2(\boldsymbol{R}\boldsymbol{R}^\top \otimes \boldsymbol{W}\boldsymbol{W}^*))^{-1}\\ &= \sigma^{-2}\boldsymbol{I} - \sigma^{-2}\sigma_b^2 \underbrace{(\boldsymbol{U}_K\Sigma_K \otimes \boldsymbol{U}_W\Sigma_W)}_{DT\times KS}\\ &\times \underbrace{(\sigma^2\boldsymbol{I} + \sigma_b^2(\Sigma_K^2 \otimes \Sigma_W^2))^{-1}}_{KS\times KS\text{ (diagonal)}} \underbrace{(\boldsymbol{U}_K\Sigma_K \otimes \boldsymbol{U}_W\Sigma_W)^*}_{KS\times DT}.\end{aligned}$$

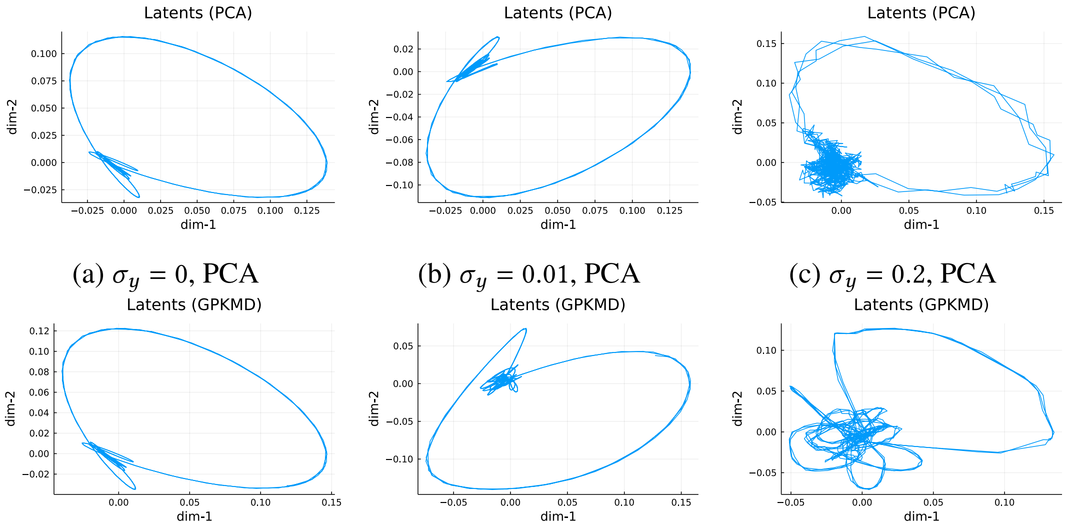
On the other hand, the log det(·) of the covariance matrix can be transformed by the Weinstein–Aronszajn identity [Katō, 1995],

$$\begin{aligned}&\log\det(\sigma^2\boldsymbol{I} + \sigma_b^2(\boldsymbol{K}_1 \otimes \boldsymbol{W}\boldsymbol{W}^*))\\ &\approx \log\det(\sigma^2\boldsymbol{I} + \sigma_b^2(\boldsymbol{R}\boldsymbol{R}^\top \otimes \boldsymbol{W}\boldsymbol{W}^*))\\ &= (DT - KS)\log\sigma^2 + \log\det\underbrace{(\sigma^2\boldsymbol{I} + \sigma_b^2(\Sigma_K^2 \otimes \Sigma_W^2))}_{KS\times KS\text{ (diagonal)}}.\end{aligned}$$

Since the computational complexity of our approach is dominated by the Nyström method (or incomplete Cholesky decomposition) and SVD, it is markedly reduced to $O(DK^2 + TS^2)$ for $K \ll D$ and $S \ll T$. Notably, it is unnecessary to store the $T \times T$ matrix $\boldsymbol{K}_1$ (and $\boldsymbol{K}_0$) in memory in both the Nyström and incomplete Cholesky decomposition algorithms. Therefore, the space complexity of GPKMD can be reduced to $O(DK + TS)$. The gradients of GPKMD can also be evaluated in a short time, as shown in Section 3.8.

3.5 Experiments

In this section, we demonstrate GPKMD in two experimental settings, one with a synthetic dataset and one with a real-world dataset. Through the experiments below, we show that a wide range of information about given data is available from the estimated parameters of GPKMD. We employed MAP estimation by the conjugate gradient method



(a) $\sigma_y = 0$, PCA

(b) $\sigma_y = 0.01$, PCA

(c) $\sigma_y = 0.2$, PCA

(d) $\sigma_y = 0$, GPKMD

(e) $\sigma_y = 0.01$, GPKMD

(f) $\sigma_y = 0.2$, GPKMD

Figure 3.1: Latent variables estimated by PCA and GPKMD for $P = 2$ and different noise levels, $\sigma_y = 0, 0.01, 0.2$.

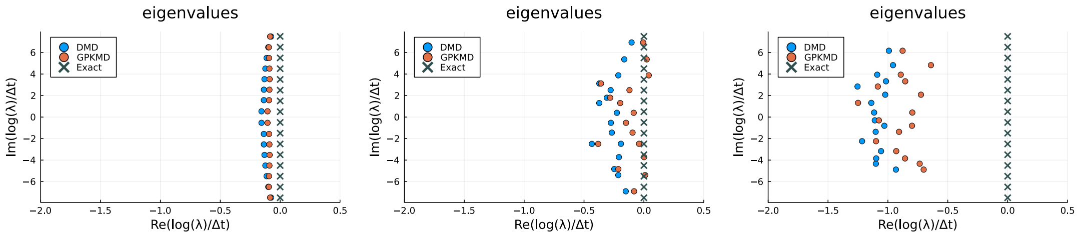
for learning GPKMD. The estimation of GPKMD parameters is sensitive to the initial values since the posterior defined with (3.12) and (3.13) is non-convex. For the initial values of GPKMD, we used PCA results for the latent variables $\boldsymbol{X}$ and standard DMD results for the Koopman eigenvalues $\{\lambda_k\}$ and modes $\{\boldsymbol{w}_k\}$. For the kernel functions of GPKMD, we employed an RBF kernel for $k(\cdot,\cdot)$ in (3.12) and an RBF+linear kernel for $k_x(\cdot,\cdot)$ in (3.13).

3.5.1 Stuart–Landau Equation

First, we applied GPKMD to a synthetic dataset that follows the Stuart–Landau equation. The Stuart–Landau equation is a well-known nonlinear dynamical system that has the discretized form

$$r_{t+1} = r_t + (\delta r_t - r_t^3)\Delta t$$
$$\theta_{t+1} = \theta_t + (\gamma - \beta r_t^2)\Delta t$$

in polar coordinates. The behavior of the system is determined by the parameters δ, γ, and β. For example, $\delta > 0$ induces the limit cycle.



(a) $\sigma = 0$ (b) $\sigma = 0.01$ (c) $\sigma = 0.2$

Figure 3.2: Eigenvalues $\{\lambda_k^{\text{cont}}\}$ estimated by DMD and GPKMD.

Table 3.1: Absolute errors of the estimated eigenvalues $\|\text{Re}(\boldsymbol{\lambda}^{\text{exact}} - \boldsymbol{\lambda}^{\text{cont}})\|$.

	$\sigma = 0$	$\sigma = 0.01$	$\sigma = 0.2$
DMD	0.49	1.06	4.32
GPKMD	0.37	0.71	3.61

We generated data with $\delta = 0.5, \beta = \gamma = 1, \Delta t = 0.05$, and data length $T = 751$. As the observed data $Y = (y_{dt})$ obtained through an observable, we employed

$$
\begin{aligned}
y_{dt} &= g_d(r_t, \theta_t) + \epsilon_{dt} = \exp(id'\theta_t) + \epsilon_{dt} \\
d' &= \begin{cases} d' = d - \lceil D/2 \rceil & (d \text{ is odd}) \\ d' = d/2 & (d \ \text{ is even}) \end{cases} \\
\epsilon_{dt} &\sim \mathcal{CN}(0, \sigma_y^2),
\end{aligned}
$$

with input dimension $D = 35$ and noise levels $\sigma_y = 0, 0.01, 0.2$. We used $K = 16$ modes, $P = 2$ latent dimensions, and $S = 50$ as the rank of the Gram matrices. Figure 3.1 shows the latent variables estimated by PCA and GPKMD. Although PCA and GPKMD estimates nearly the same trajectories for $\sigma_y = 0, 0.01$, at the higher noise level $\sigma_y = 0.2$, the latent variables of PCA are buried in the noise around the origin $\boldsymbol{x}_t = (0, 0)^\top$. In contrast, GPKMD captures a contiguous and periodic trajectory around the origin for $\sigma_y = 0.2$. The estimated Koopman eigenvalues corresponding to the continuous system $\lambda_k^{\text{cont}} = \log(\lambda_k)/\Delta t$ are shown in Figure 3.2. Note that the exact eigenvalues of the continuous system are known:

$$\lambda_{ln}^{\text{exact}} = -2l\delta + in\omega_0,$$
$$\omega_0 = \gamma - \beta\delta,$$

where $l \in \mathbb{N}$ and $n \in \mathbb{Z}$ [Črnjarić-Žic et al., 2020]. As seen in Figure 3.2 and Table 3.1 [2], GPKMD estimates the Koopman eigenvalues more accurately than DMD. The estimates of DMD tend to shrink as the noise level increases. Meanwhile, though depending on initial values and hyperparameters, GPKMD is more robust than DMD for this dynamical system.

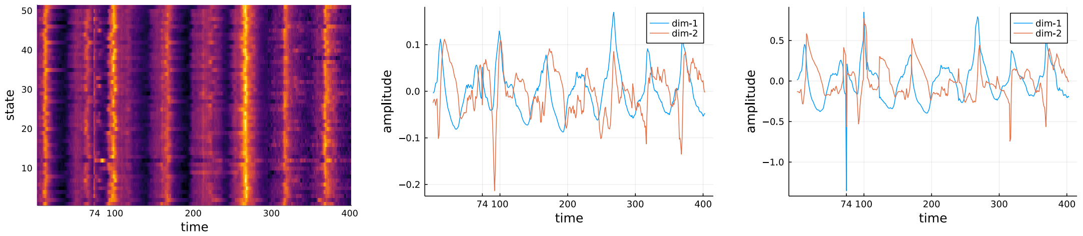
3.5.2 Google Flu Trends

Google has attempted to predict weekly spatiotemporal flu activity from query data from its search engine. The project Google Flu Trends has been discontinued, but the predicted results are available[3]. Proctor and Eckhoff [2015] analyzed the Google Flu Trends data by DMD, and we take a similar approach here. We focus on the values in the US and extracted the interval from 2007–12–02 to 2015–08–09 to avoid missing data, so that the input size was $D = 51, T = 402$. Considering the nature of the data, we applied log-transformation before statewise standardization as preprocessing. In the preprocessed input shown in Figure 3.3a, a rough periodicity can be observed. We set $K = 6$ modes, $P = 2$ latent dimensions, and $S = 50$ as the rank of the Gram matrices.

Figures 3.3b and 3.3c show the latent variables estimated by PCA and GPKMD, respectively. Note that the latent variables estimated by PCA are used as the initial values of those estimated by GPKMD. The latent variables estimated by GPKMD clearly show anomalous behavior at $t = 74$, unlike those estimated by PCA. An anomalous spike at $t = 74$ can also be observed in the original input (Figure 3.3a), but it does not appear to be outlying in the sense of i.i.d. observation. $t = 74$ indicates the period between 2009–04–26 to 2009–05–02. At the time, interestingly, the US was in turmoil due to the

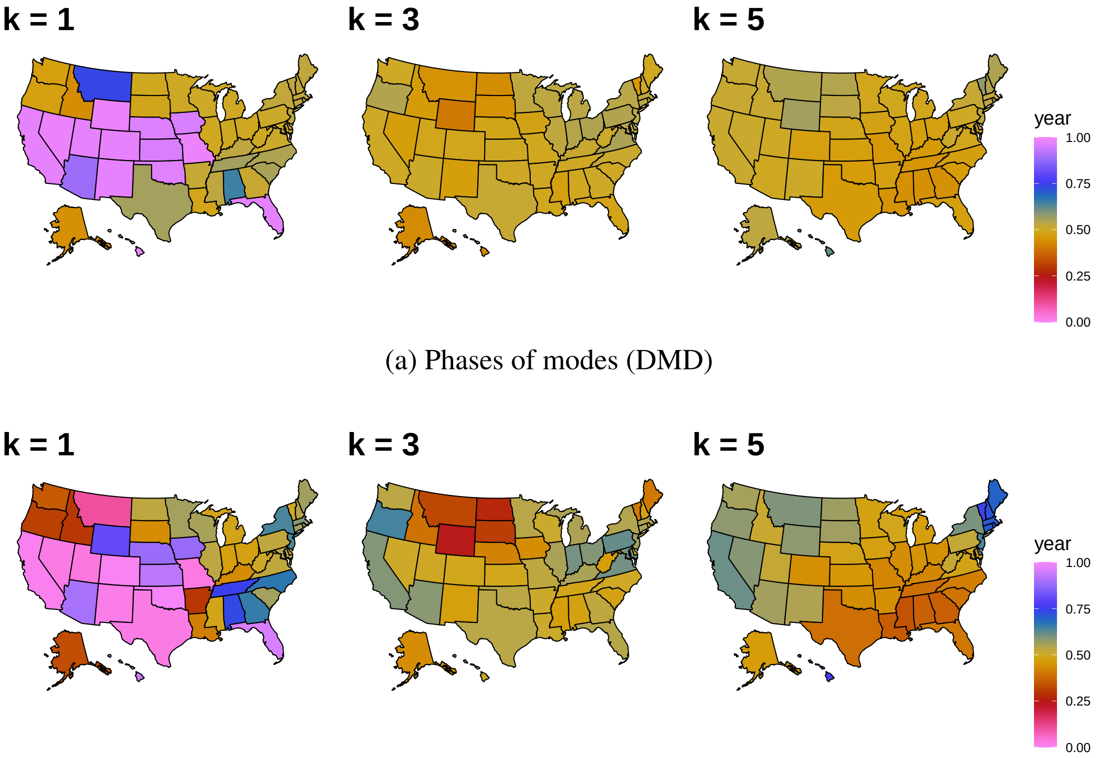
[2] Because $\text{Im}(\lambda_k^{\text{cont}})$ is equal for DMD and GPKMD in our setting, as discussed in Section 3.6, we only consider the real parts to obtain the errors.

[3] The estimates can be accessed at https://www.google.com/publicdata/explore?ds=z3bsqef7ki44ac_&hl=en&dl=en, and the raw data is archived at http://web.archive.org/web/*/http://www.google.org/flutrends/.



(a) Input (b) Latents (PCA) (c) Latents (GPKMD)

Figure 3.3: (a) Input from Google Flu Trends in the US. (b) Latent variables estimated by PCA. (c) Latent variables estimated by GPKMD.



(a) Phases of modes (DMD)

(b) Phases of modes (GPKMD)

Figure 3.4: Phases of 1, 3, and 5-th modes estimated by (a) DMD and (b) GPKMD. Each phase indicates the time of a year between 0 and 1.

pandemic by the new influenza A (H1N1). In fact, WHO has raised the level of influenza pandemic alert to phase 4 on 2009–04–27, and again raised to phase 5 on 2009–04–29 [World Health Organization, 2013]. The spike may reflect this social situation. It is considered that the temporal structure and nonlinearity of GPKMD increase the sensitivity to such temporally anomalous behavior. In addition, the estimated modes $\{\boldsymbol{w}_k\}$ provide information about the phase shifts, that is, the phase of the k-th mode in the d-th state is computed from $\arg w_{dk}$. Suppose that $\arg w_{dk}$ is wrapped to $[0, 2\pi)$, then $\arg w_{dk} \,/\, 2\pi \in [0, 1)$ expresses the shift within a year. Figure 3.4 shows the phases of the modes corresponding to the indices $k = 1, 3, 5$, estimated by DMD and GPKMD. The modes indexed by even numbers are omitted because they have conjugate elements of odd numbers. The first modes of DMD and GPKMD indicate some state clusters. We also find a clustered relationship in the northern states at $k = 3$ and a gradual slope from the southeast to the northwest at $k = 5$. Such phase structures are considered to reflect seasonal transitions of flu trends. Notably a similar smooth phase transition of a dynamic mode has also been reported in a previous work [Proctor and Eckhoff, 2015], but the transition is more pronounced for our method.

3.6 Discussion

In this chapter, we developed a nonlinear probabilistic generative model of KMD based on unsupervised GP, and we also proposed its efficient inference scheme via low-rank approximations of covariance matrices. Our method, named GPKMD, is advantageous in terms of the comprehensiveness of the parameter set to be estimated. Since each quantity in KMD (3.4) is physically meaningful, the comprehensiveness of GPKMD directly means that rich information can be obtained. We also examined the scalability of GPKMD in Section 3.4. By exploiting the properties of the Kronecker product and low-rank approximations of matrices, we markedly reduced the computational complexity from $O(D^3 + T^3)$ to $O(DK^2 + TS^2)$, where $K \ll D$ and $S \ll T$.

3.7 Derivation of the Marginal Likelihood

Properties of Kronecker Product and Vec Operator

We introduce some properties of the Kronecker product and vec operator for simplicity in the calculations below:

$$
\begin{aligned}
(\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}) &= (\boldsymbol{AC}) \otimes (\boldsymbol{BD}), && (3.14)\\
\mathrm{vec}(\boldsymbol{ABC}) &= (\boldsymbol{C}^\top \otimes \boldsymbol{A})\mathrm{vec}(\boldsymbol{B}), && (3.15)\\
\mathrm{tr}(\boldsymbol{A} \otimes \boldsymbol{B}) &= \mathrm{tr}(\boldsymbol{A})\mathrm{tr}(\boldsymbol{B}), && (3.16)\\
\mathrm{vec}(\boldsymbol{A})^* \mathrm{vec}(\boldsymbol{B}) &= \mathrm{tr}(\boldsymbol{A}^* \boldsymbol{B}).
\end{aligned}
$$

Derivation of the Marginal Likelihood

From each $\mathcal{CN}(\cdot, \cdot)$ in (3.11) and the prior $p(b_{kl}) = \mathcal{CN}(0, \sigma_b^2)$, we can marginalize out the coefficients $\{b_{kl}\}$ analytically. Considering the joint marginal likelihood for the first $\mathcal{CN}(\cdot, \cdot)$ in (3.11), we have

$$
\begin{aligned}
p(Y|\boldsymbol{X}, \boldsymbol{W}, \sigma^2) &= \int \prod_{t=1}^{T} p(\boldsymbol{y}_t | \{\boldsymbol{x}_t\}, \{\lambda_k\}, \{\boldsymbol{w}_k\}, \{b_{kl}\}, \sigma^2) \cdot \prod_{k,l} p(b_{kl}) db_{kl} \\
&= \int \mathcal{CN}(\mathrm{vec}(Y) | \mathrm{vec}(\boldsymbol{WB\Psi}_1), \sigma^2 \boldsymbol{I}) \mathcal{CN}(\mathrm{vec}(\boldsymbol{B}) | \boldsymbol{0}, \sigma_b^2 \boldsymbol{I}) d\boldsymbol{B}, \quad (3.17)
\end{aligned}
$$

where $\boldsymbol{\Psi}_1 = (\boldsymbol{\psi}(\boldsymbol{x}_1), \boldsymbol{\psi}(\boldsymbol{x}_2), \ldots)$ and $\boldsymbol{B}$ is the matrix whose (k, l)-th element is b_{kl}. Using the relations (3.14) and (3.15), we find that the integrand in (3.17) is proportional to

$$
\begin{aligned}
&\mathcal{CN}(\mathrm{vec}(Y) | \mathrm{vec}(\boldsymbol{WB\Psi}_1), \sigma^2 \boldsymbol{I}) \mathcal{CN}(\mathrm{vec}(\boldsymbol{B}) | \boldsymbol{0}, \sigma_b^2 \boldsymbol{I}) \\
&\quad \propto \exp\left(-\sigma^{-2} \|\mathrm{vec}(Y) - \mathrm{vec}(\boldsymbol{WB\Psi}_1)\|^2 - \sigma_b^2 \|\mathrm{vec}(\boldsymbol{B})\|^2\right) \\
&\quad = \exp\left\{-\sigma^{-2}\left(\|\mathrm{vec}(Y)\|^2 + \|\mathrm{vec}(\boldsymbol{B}) + \mathrm{vec}(\bar{\boldsymbol{B}})\|_{\Sigma_B^{-1}}^2 - \|\mathrm{vec}(\bar{\boldsymbol{B}})\|_{\Sigma_B^{-1}}^2\right)\right\}, \quad (3.18)
\end{aligned}
$$

where

$$\begin{aligned}
\Sigma_B^{-1} &= (\boldsymbol{\Psi}_1\boldsymbol{\Psi}_1^\top) \otimes (\boldsymbol{W}^*\boldsymbol{W}) + \sigma^2\sigma_b^{-2}\boldsymbol{I}, \\
\mathrm{vec}(\bar{\boldsymbol{B}}) &= \Sigma_B \mathrm{vec}(\boldsymbol{W}^*\boldsymbol{Y}\boldsymbol{\Psi}_1^\top), \\
\|\boldsymbol{z}\|_{\Sigma_B^{-1}}^2 &= \boldsymbol{z}^*\Sigma_B^{-1}\boldsymbol{z}.
\end{aligned}$$

Since (3.18) is the squared exponential form w.r.t. vec($\boldsymbol{B}$), the integral (3.17) can be evaluated as

$$\begin{aligned}
&\int \mathcal{CN}(\mathrm{vec}(\boldsymbol{Y})|\mathrm{vec}(\boldsymbol{W}\boldsymbol{B}\boldsymbol{\Psi}_1), \sigma^2\boldsymbol{I})\mathcal{CN}(\mathrm{vec}(\boldsymbol{B})|\boldsymbol{0}, \sigma_b^2\boldsymbol{I})d\boldsymbol{B} \\
&\propto \exp\left\{-\sigma^{-2}\left(\|\mathrm{vec}(\boldsymbol{Y})\|^2 - \|\mathrm{vec}(\bar{\boldsymbol{B}})\|_{\Sigma_B^{-1}}^2\right)\right\},
\end{aligned} \tag{3.19}$$

and this should also be Gaussian w.r.t. vec($\boldsymbol{Y}$). Here, applying the Woodbury identity and (3.15), we obtain

$$\begin{aligned}
\Sigma_B &= ((\boldsymbol{\Psi}_1\boldsymbol{\Psi}_1^\top) \otimes (\boldsymbol{W}^*\boldsymbol{W}) + \sigma^2\sigma_b^{-2}\boldsymbol{I})^{-1} \\
&= \sigma^{-2}\sigma_b^2\{\boldsymbol{I} - \sigma_b^2(\boldsymbol{\Psi}_1 \otimes \boldsymbol{W}^*)\Sigma_Y^{-1}(\boldsymbol{\Psi}_1^\top \otimes \boldsymbol{W})\},
\end{aligned}$$

where we define

$$\begin{aligned}
\Sigma_Y &= \sigma^2\boldsymbol{I} + \sigma_b^2\boldsymbol{K}_1 \otimes (\boldsymbol{W}\boldsymbol{W}^*), \\
\boldsymbol{K}_1 &= \boldsymbol{\Psi}_1^\top\boldsymbol{\Psi}_1.
\end{aligned}$$

Then, $\|\mathrm{vec}(\bar{\boldsymbol{B}})\|^2_{\Sigma_B^{-1}}$ in (3.19) can be simplified to

$$
\begin{aligned}
&\|\mathrm{vec}(\bar{\boldsymbol{B}})\|^2_{\Sigma_B^{-1}} \\
&\quad = \mathrm{vec}(\boldsymbol{W}^*\boldsymbol{Y}\boldsymbol{\Psi}_1^\top)^*\Sigma_B\mathrm{vec}(\boldsymbol{W}^*\boldsymbol{Y}\boldsymbol{\Psi}_1^\top) \\
&\quad = \sigma^{-2}\sigma_b^2\|\mathrm{vec}(\boldsymbol{W}^*\boldsymbol{Y}\boldsymbol{\Psi}_1^\top)\|^2 - \sigma^{-2}\sigma_b^2\|(\boldsymbol{\Psi}_1^\top\otimes\boldsymbol{W})\mathrm{vec}(\boldsymbol{W}^*\boldsymbol{Y}\boldsymbol{\Psi}_1^\top)\|^2_{\sigma_b^2\Sigma_Y^{-1}} \\
&\quad = \sigma^{-2}\sigma_b^2\mathrm{tr}(\boldsymbol{\Psi}_1\boldsymbol{Y}^*\boldsymbol{W}\boldsymbol{W}^*\boldsymbol{Y}\boldsymbol{\Psi}_1^\top) - \sigma^{-2}\sigma_b^2\|\mathrm{vec}(\boldsymbol{W}\boldsymbol{W}^*\boldsymbol{Y}\boldsymbol{K}_1)\|^2_{\sigma_b^2\Sigma_Y^{-1}} \\
&\quad = \sigma^{-2}\sigma_b^2\mathrm{vec}(\boldsymbol{W}\boldsymbol{W}^*\boldsymbol{Y})^*\mathrm{vec}(\boldsymbol{Y}\boldsymbol{K}_1) - \sigma^{-2}\sigma_b^2\|(\boldsymbol{K}_1\otimes(\boldsymbol{W}\boldsymbol{W}^*))\mathrm{vec}(\boldsymbol{Y})\|^2_{\sigma_b^2\Sigma_Y^{-1}} \\
&\quad = \sigma^{-2}\sigma_b^2\mathrm{vec}(\boldsymbol{Y})^* \\
&\qquad \times [\boldsymbol{K}_1\otimes(\boldsymbol{W}\boldsymbol{W}^*) - \sigma_b^2(\boldsymbol{K}_1\otimes(\boldsymbol{W}\boldsymbol{W}^*))\Sigma_Y^{-1}(\boldsymbol{K}_1\otimes(\boldsymbol{W}\boldsymbol{W}^*))]\mathrm{vec}(\boldsymbol{Y}) \\
&\quad = \mathrm{vec}(\boldsymbol{Y})^*(\boldsymbol{I} - \sigma^2\Sigma_Y^{-1})\mathrm{vec}(\boldsymbol{Y}),
\end{aligned}
$$

where we use the exact relation $\boldsymbol{A} - \boldsymbol{A}(\boldsymbol{A}+\boldsymbol{B})^{-1}\boldsymbol{A} = \boldsymbol{B} - \boldsymbol{B}(\boldsymbol{A}+\boldsymbol{B})^{-1}\boldsymbol{B}$ for the rightmost transform. Now, the (unnormalized) marginal likelihood (3.19) becomes

$$
\begin{aligned}
p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{W},\sigma^2) &\propto \exp\left\{-\sigma^{-2}\left(\|\mathrm{vec}(\boldsymbol{Y})\|^2 - \|\mathrm{vec}(\bar{\boldsymbol{B}})\|^2_{\Sigma_B^{-1}}\right)\right\} \\
&= \exp\left\{-\sigma^{-2}\left(\mathrm{vec}(\boldsymbol{Y})^*\mathrm{vec}(\boldsymbol{Y}) - \mathrm{vec}(\boldsymbol{Y})^*(\boldsymbol{I} - \sigma^2\Sigma_Y^{-1})\mathrm{vec}(\boldsymbol{Y})\right)\right\} \\
&= \exp\left(-\mathrm{vec}(\boldsymbol{Y})^*\Sigma_Y^{-1}\mathrm{vec}(\boldsymbol{Y})\right),
\end{aligned}
$$

so that $p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{W},\sigma^2,\sigma_b^2) = \mathcal{CN}(\mathrm{vec}(\boldsymbol{Y})|\boldsymbol{0},\Sigma_Y)$. Applying a similar manner to the second $\mathcal{CN}(\cdot,\cdot)$ in (3.11), we can finally obtain the marginal likelihood of GPKMD (3.12).

3.8 Derivatives of the Marginal Likelihood

In Section 3.4, we show that the low-rank approximations for the covariance matrices reduce the computational cost of evaluating the GPKMD likelihood. Similarly, we can evaluate derivatives of the likelihood efficiently. For the complex-valued parameters of GPKMD, we define the complex gradient of $f : \mathbb{C}^D \to \mathbb{R}$ w.r.t. $\boldsymbol{\theta} \in \mathbb{C}^D$ as

$$
\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) = \frac{\partial f(\boldsymbol{\theta})}{\partial \mathrm{Re}(\boldsymbol{\theta})} + i\frac{\partial f(\boldsymbol{\theta})}{\partial \mathrm{Im}(\boldsymbol{\theta})}.
$$

In general, we consider the cost function

$$\begin{aligned}\ell(\boldsymbol{\theta}^g, \boldsymbol{\theta}^h) &= \log\det(\sigma^2 \boldsymbol{I} + \boldsymbol{G}(\boldsymbol{\theta}^g) \otimes \boldsymbol{H}(\boldsymbol{\theta}^h)) \\ &\quad - \mathrm{vec}(Y)^*(\sigma^2 \boldsymbol{I} + \boldsymbol{G}(\boldsymbol{\theta}^g) \otimes \boldsymbol{H}(\boldsymbol{\theta}^h))^{-1}\mathrm{vec}(Y),\end{aligned}$$

where $\boldsymbol{G}$ and $\boldsymbol{H}$ are positive semidefinite and $\boldsymbol{\theta}^g$ and $\boldsymbol{\theta}^h$ are the parameter vectors to be learned. As introduced in Section 3.4, suppose that we obtain low-rank representations such that $\boldsymbol{G} \approx \boldsymbol{U}_G\Sigma_G^2\boldsymbol{U}_G^*$ and $\boldsymbol{H} \approx \boldsymbol{U}_H\Sigma_H^2\boldsymbol{U}_H^*$ by SVD. Then, the Woodbury identity enables the following approximation:

$$\begin{aligned}&(\sigma^2 \boldsymbol{I} + \boldsymbol{G} \otimes \boldsymbol{H})^{-1} \\ &\quad \approx \sigma^{-2}\{\boldsymbol{I} - [(\boldsymbol{U}_G\Sigma_G) \otimes (\boldsymbol{U}_H\Sigma_H)](\sigma^2\boldsymbol{I} + \Sigma_G^2 \otimes \Sigma_H^2)^{-1}[(\boldsymbol{U}_G\Sigma_G) \otimes (\boldsymbol{U}_H\Sigma_H)]^*\}.\end{aligned}$$

Derivatives w.r.t. $\boldsymbol{\theta}^g$

The derivative of the cost function $\ell(\boldsymbol{\theta}^g, \boldsymbol{\theta}^h)$ w.r.t. θ_i^g is

$$\begin{aligned}\nabla_{\theta_i^g}\ell &= -\mathrm{tr}\{(\sigma^2\boldsymbol{I} + \boldsymbol{G} \otimes \boldsymbol{H})^{-1}(\nabla_{\theta_i^g}\boldsymbol{G} \otimes \boldsymbol{H})\} \\ &\quad + \mathrm{vec}(Y)^*(\sigma^2\boldsymbol{I} + \boldsymbol{G} \otimes \boldsymbol{H})^{-1}(\nabla_{\theta_i^g}\boldsymbol{G} \otimes \boldsymbol{H})(\sigma^2\boldsymbol{I} + \boldsymbol{G} \otimes \boldsymbol{H})^{-1}\mathrm{vec}(Y). \qquad (3.20)\end{aligned}$$

The first term in (3.20) can be approximated by

$$\begin{aligned}&\mathrm{tr}\{(\sigma^2\boldsymbol{I} + \boldsymbol{G} \otimes \boldsymbol{H})^{-1}(\nabla_{\theta_i^g}\boldsymbol{G} \otimes \boldsymbol{H})\} \\ &\quad \approx \sigma^{-2}\mathrm{tr}(\nabla_{\theta_i^g}\boldsymbol{G} \otimes \boldsymbol{H}) - \sigma^{-2}\mathrm{tr}\{[(\boldsymbol{U}_G\Sigma_G) \otimes (\boldsymbol{U}_H\Sigma_H)] \\ &\qquad \times (\sigma^2\boldsymbol{I} + \Sigma_G^2 \otimes \Sigma_H^2)^{-1}[(\boldsymbol{U}_G\Sigma_G) \otimes (\boldsymbol{U}_H\Sigma_H)]^*(\nabla_{\theta_i^g}\boldsymbol{G} \otimes \boldsymbol{H})\} \\ &\quad = \sigma^{-2}\mathrm{tr}(\nabla_{\theta_i^g}\boldsymbol{G})\mathrm{tr}(\boldsymbol{H}) - \sigma^{-2}\mathrm{tr}\{(\sigma^2\boldsymbol{I} + \Sigma_G^2 \otimes \Sigma_H^2)^{-1} \\ &\qquad \times [(\Sigma_G\boldsymbol{U}_G^*\nabla_{\theta_i^g}\boldsymbol{G}\boldsymbol{U}_G\Sigma_G) \otimes \Sigma_H^4]\} \\ &\quad = \sigma^{-2}\mathrm{tr}(\nabla_{\theta_i^g}\boldsymbol{G})\mathrm{tr}(\Sigma_H^2) \\ &\qquad - \sigma^{-2}\mathrm{diag}\{(\sigma^2\boldsymbol{I} + \Sigma_G^2 \otimes \Sigma_H^2)^{-1}\}^\top\mathrm{diag}\{(\Sigma_G\boldsymbol{U}_G^*\nabla_{\theta_i^g}\boldsymbol{G}\boldsymbol{U}_G\Sigma_G) \otimes \Sigma_H^4\} \\ &\quad = \sigma^{-2}\mathrm{tr}(\nabla_{\theta_i^g}\boldsymbol{G})\mathrm{tr}(\Sigma_H^2) \\ &\qquad - \sigma^{-2}\mathrm{diag}\{(\sigma^2\boldsymbol{I} + \Sigma_G^2 \otimes \Sigma_H^2)^{-1}\}^\top\{\mathrm{diag}(\Sigma_G\boldsymbol{U}_G^*\nabla_{\theta_i^g}\boldsymbol{G}\boldsymbol{U}_G\Sigma_G) \otimes \mathrm{diag}(\Sigma_H^4)\},\end{aligned}$$

where we use (3.16) and $\mathrm{tr}(\boldsymbol{D}\boldsymbol{A}) = \mathrm{diag}(\boldsymbol{D})^\top \mathrm{diag}(\boldsymbol{A})$ for any diagonal matrix $\boldsymbol{D}$. Note that if $\boldsymbol{G}$ is a Gram matrix of latent variables $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T)^\top$, i.e., $\boldsymbol{G} = (k(\boldsymbol{x}_i, \boldsymbol{x}_j))_{ij}$ $(= \boldsymbol{K}_1$ in (3.12)), then the elements of $\nabla_{x_{pi}}\boldsymbol{G}$ become zeros except for the i-th row and column, and $\mathrm{tr}(\nabla_{x_{pi}}\boldsymbol{G}) = 0$. In such a case, further simplification is possible:

$$\begin{aligned}
&\mathrm{tr}\{(\sigma^2\boldsymbol{I} + \boldsymbol{G}\otimes\boldsymbol{H})^{-1}(\nabla_{x_{pi}}\boldsymbol{G}\otimes\boldsymbol{H})\}\\
&\quad\approx -2\sigma^{-2}\mathrm{diag}\{(\sigma^2\boldsymbol{I} + \Sigma_G^2\otimes\Sigma_H^2)^{-1}\}^\top\\
&\qquad\times\{[(\boldsymbol{U}_G^*\nabla_{x_{pi}}\boldsymbol{G}_{:i})\odot\boldsymbol{U}_{G,:i}\odot\mathrm{diag}(\Sigma_G^2)]\otimes\mathrm{diag}(\Sigma_H^4)\},
\end{aligned}$$

where $\odot$ denotes the Hadamard product and $\nabla_{x_{pi}}\boldsymbol{G}_{:i}$ and $\boldsymbol{U}_{G,:i}$ are the i-th column vectors of $\nabla_{x_{pi}}\boldsymbol{G}$ and $\boldsymbol{U}_G$, respectively.

We next consider the second term in (3.20). By defining the transformed data onto the lower dimension

$$\mathrm{vec}(\tilde{\boldsymbol{Y}}) = (\sigma^2\boldsymbol{I} + \Sigma_G^2\otimes\Sigma_H^2)^{-1}\mathrm{vec}(\Sigma_H\boldsymbol{U}_H^*\boldsymbol{Y}\boldsymbol{U}_G\Sigma_G),$$

we obtain the following approximation:

$$\begin{aligned}
&\mathrm{vec}(\boldsymbol{Y})^*(\sigma^2\boldsymbol{I} + \boldsymbol{G}\otimes\boldsymbol{H})^{-1}(\nabla_{\theta_i^g}\boldsymbol{G}\otimes\boldsymbol{H})(\sigma^2\boldsymbol{I} + \boldsymbol{G}\otimes\boldsymbol{H})^{-1}\mathrm{vec}(\boldsymbol{Y})\\
&\quad\approx\sigma^{-4}\{\mathrm{vec}(\boldsymbol{Y}) - [(\boldsymbol{U}_G\Sigma_G)\otimes(\boldsymbol{U}_H\Sigma_H)]\mathrm{vec}(\tilde{\boldsymbol{Y}})\}^*[\nabla_{\theta_i^g}\boldsymbol{G}\otimes(\boldsymbol{U}_H\Sigma_H^2\boldsymbol{U}_H^*)]\\
&\qquad\times\{\mathrm{vec}(\boldsymbol{Y}) - [(\boldsymbol{U}_G\Sigma_G)\otimes(\boldsymbol{U}_H\Sigma_H)]\mathrm{vec}(\tilde{\boldsymbol{Y}})\}\\
&\quad=\sigma^{-4}\mathrm{vec}(\boldsymbol{Y})^*[\nabla_{\theta_i^g}\boldsymbol{G}\otimes(\boldsymbol{U}_H\Sigma_H^2\boldsymbol{U}_H^*)]\mathrm{vec}(\boldsymbol{Y})\\
&\qquad-\sigma^{-4}\mathrm{vec}(\tilde{\boldsymbol{Y}})^*[(\Sigma_G\boldsymbol{U}_G^*\nabla_{\theta_i^g}\boldsymbol{G})\otimes(\Sigma_H^3\boldsymbol{U}_H^*)]\mathrm{vec}(\boldsymbol{Y})\\
&\qquad-\sigma^{-4}\mathrm{vec}(\boldsymbol{Y})^*[(\nabla_{\theta_i^g}\boldsymbol{G}\boldsymbol{U}_G\Sigma_G)\otimes(\boldsymbol{U}_H\Sigma_H^3)]\mathrm{vec}(\tilde{\boldsymbol{Y}})\\
&\qquad+\sigma^{-4}\mathrm{vec}(\tilde{\boldsymbol{Y}})^*[(\Sigma_G\boldsymbol{U}_G^*\nabla_{\theta_i^g}\boldsymbol{G}\boldsymbol{U}_G\Sigma_G)\otimes\Sigma_H^4)]\mathrm{vec}(\tilde{\boldsymbol{Y}})\\
&\quad=\sigma^{-4}\mathrm{tr}(\Sigma_H^2\boldsymbol{U}_H^*\boldsymbol{Y}\nabla_{\theta_i^g}\boldsymbol{G}^\top\boldsymbol{Y}^*\boldsymbol{U}_H) - \sigma^{-4}\mathrm{tr}(\Sigma_H^3\boldsymbol{U}_H^*\boldsymbol{Y}\nabla_{\theta_i^g}\boldsymbol{G}^\top\overline{\boldsymbol{U}_G}\Sigma_G\tilde{\boldsymbol{Y}}^*)\\
&\qquad-\sigma^{-4}\mathrm{tr}(\tilde{\boldsymbol{Y}}\Sigma_G\boldsymbol{U}_G^\top\nabla_{\theta_i^g}\boldsymbol{G}^\top\boldsymbol{Y}^*\boldsymbol{U}_H\Sigma_H^3) + \sigma^{-4}\mathrm{tr}(\Sigma_H^4\tilde{\boldsymbol{Y}}\Sigma_G\boldsymbol{U}_G^\top\nabla_{\theta_i^g}\boldsymbol{G}^\top\overline{\boldsymbol{U}_G}\Sigma_G\tilde{\boldsymbol{Y}}^*),
\end{aligned}$$

where $\overline{\boldsymbol{U}_H}$ denotes the conjugate matrix without the transpose of $\boldsymbol{U}_H$. Furthermore, in the particular case where $\boldsymbol{G}$ is a Gram matrix consisting of $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T)^\top$, we have

$$\begin{aligned}
&\mathrm{vec}(Y)^*(\sigma^2\boldsymbol{I} + \boldsymbol{G} \otimes \boldsymbol{H})^{-1}(\nabla_{x_{pi}}\boldsymbol{G} \otimes \boldsymbol{H})(\sigma^2\boldsymbol{I} + \boldsymbol{G} \otimes \boldsymbol{H})^{-1}\mathrm{vec}(Y) \\
&\quad \approx 2\sigma^{-4}\mathrm{Re}\{\nabla_{x_{pi}}\boldsymbol{G}_{:i}^\top Y^*\boldsymbol{U}_H\Sigma_H^2\boldsymbol{U}_H^*Y_{:t} - \nabla_{x_{pi}}\boldsymbol{G}_{:i}^\top\boldsymbol{U}_G\Sigma_G\tilde{Y}^*\Sigma_H^3\boldsymbol{U}_H^*Y_{:t} \\
&\qquad - \boldsymbol{U}_{G,i:}^\top\Sigma_G\tilde{Y}^*\Sigma_H^3\boldsymbol{U}_H Y\nabla_{x_{pi}}\boldsymbol{G}_{:i} + \nabla_{x_{pi}}\boldsymbol{G}_{:i}^\top\boldsymbol{U}_G\Sigma_G\tilde{Y}^*\Sigma_H^4\tilde{Y}\Sigma_G\boldsymbol{U}_{G,i:}\}.sl_eigvals_sigma0
\end{aligned}$$

Derivatives w.r.t. $\boldsymbol{\theta}^h$

The derivative of the cost function $\ell(\boldsymbol{\theta}^g, \boldsymbol{\theta}^h)$ w.r.t. θ_i^h is

$$\begin{aligned}
\nabla_{\theta_i^h}\ell = &-\mathrm{tr}\{(\sigma^2\boldsymbol{I} + \boldsymbol{G} \otimes \boldsymbol{H})^{-1}(\boldsymbol{G} \otimes \nabla_{\theta_i^h}\boldsymbol{H})\} \\
&+ \mathrm{vec}(Y)^*(\sigma^2\boldsymbol{I} + \boldsymbol{G} \otimes \boldsymbol{H})^{-1}(\boldsymbol{G} \otimes \nabla_{\theta_i^h}\boldsymbol{H})(\sigma^2\boldsymbol{I} + \boldsymbol{G} \otimes \boldsymbol{H})^{-1}\mathrm{vec}(Y). \qquad (3.21)
\end{aligned}$$

We can approximate the first term in (3.21) as

$$\begin{aligned}
&\mathrm{tr}\{(\sigma^2\boldsymbol{I} + \boldsymbol{G} \otimes \boldsymbol{H})^{-1}(\boldsymbol{G} \otimes \nabla_{\theta_i^h}\boldsymbol{H})\} \\
&\quad \approx \sigma^{-2}\mathrm{tr}(\boldsymbol{G} \otimes \nabla_{\theta_i^h}\boldsymbol{H}) - \sigma^{-2}\mathrm{tr}\{[(\boldsymbol{U}_G\Sigma_G) \otimes (\boldsymbol{U}_H\Sigma_H)] \\
&\qquad \times (\sigma^2\boldsymbol{I} + \Sigma_G^2 \otimes \Sigma_H^2)^{-1}[(\boldsymbol{U}_G\Sigma_G) \otimes (\boldsymbol{U}_H\Sigma_H)]^*(\boldsymbol{G} \otimes \nabla_{\theta_i^h}\boldsymbol{H})\} \\
&\quad = \sigma^{-2}\mathrm{tr}(\boldsymbol{G})\mathrm{tr}(\nabla_{\theta_i^h}\boldsymbol{H}) - \sigma^{-2}\mathrm{tr}\{(\sigma^2\boldsymbol{I} + \Sigma_G^2 \otimes \Sigma_H^2)^{-1} \\
&\qquad \times [\Sigma_G^4 \otimes (\Sigma_H\boldsymbol{U}_H^*\nabla_{\theta_i^h}\boldsymbol{H}\boldsymbol{U}_H\Sigma_H)]\} \\
&\quad = \sigma^{-2}\mathrm{tr}(\Sigma_G^2)\mathrm{tr}(\nabla_{\theta_i^h}\boldsymbol{H}) \\
&\qquad - \sigma^{-2}\mathrm{diag}\{(\sigma^2\boldsymbol{I} + \Sigma_G^2 \otimes \Sigma_H^2)^{-1}\}^\top\mathrm{diag}\{\Sigma_G^4 \otimes (\Sigma_H\boldsymbol{U}_H^*\nabla_{\theta_i^h}\boldsymbol{H}\boldsymbol{U}_H\Sigma_H)\} \\
&\quad = \sigma^{-2}\mathrm{tr}(\Sigma_G^2)\mathrm{tr}(\nabla_{\theta_i^h}\boldsymbol{H}) \\
&\qquad - \sigma^{-2}\mathrm{diag}\{(\sigma^2\boldsymbol{I} + \Sigma_G^2 \otimes \Sigma_H^2)^{-1}\}^\top\{\mathrm{diag}(\Sigma_G^4) \otimes \mathrm{diag}(\Sigma_H\boldsymbol{U}_H^*\nabla_{\theta_i^h}\boldsymbol{H}\boldsymbol{U}_H\Sigma_H)\}.
\end{aligned}$$

Consider $\boldsymbol{H} = \boldsymbol{W}\boldsymbol{W}^*$ and the derivative with respect to w_{dk} in the first $\mathcal{CN}(\cdot,\cdot)$ in (3.12). In this case, a more computationally inexpensive form is available:

$$\begin{aligned}
&\mathrm{tr}\{(\sigma^2\boldsymbol{I} + \boldsymbol{G} \otimes \boldsymbol{H})^{-1}(\boldsymbol{G} \otimes \nabla_{w_{dk}}\boldsymbol{H})\} \\
&\quad \approx 2\sigma^{-2}\mathrm{tr}(\Sigma_G^2) - 2\sigma^{-2}\mathrm{diag}\{(\sigma^2\boldsymbol{I} + \Sigma_G^2 \otimes \Sigma_H^2)^{-1}\}^\top \\
&\qquad \times \{\mathrm{diag}(\Sigma_G^4) \otimes [(\boldsymbol{U}_H^*\boldsymbol{w}_k) \odot \boldsymbol{U}_{H,d:} \odot \mathrm{diag}(\Sigma_H^2)]\}.
\end{aligned}$$

For the second term in (3.21), the following approximation is similarly obtained:

$$\begin{aligned}
&\mathrm{vec}(Y)^*(\sigma^2\boldsymbol{I}+\boldsymbol{G}\otimes\boldsymbol{H})^{-1}(\boldsymbol{G}\otimes\nabla_{\theta_i^h}\boldsymbol{H})(\sigma^2\boldsymbol{I}+\boldsymbol{G}\otimes\boldsymbol{H})^{-1}\mathrm{vec}(Y)\\
&\quad\approx\sigma^{-4}\{\mathrm{vec}(Y)-[(\boldsymbol{U}_G\Sigma_G)\otimes(\boldsymbol{U}_H\Sigma_H)]\mathrm{vec}(\tilde{Y})\}^*[(\boldsymbol{U}_G\Sigma_G^2\boldsymbol{U}_G^*)\otimes\nabla_{\theta_i^h}\boldsymbol{H}]\\
&\qquad\times\{\mathrm{vec}(Y)-[(\boldsymbol{U}_G\Sigma_G)\otimes(\boldsymbol{U}_H\Sigma_H)]\mathrm{vec}(\tilde{Y})\}\\
&\quad=\sigma^{-4}\mathrm{vec}(Y)^*[(\boldsymbol{U}_G\Sigma_G^2\boldsymbol{U}_G^*)\otimes\nabla_{\theta_i^h}\boldsymbol{H}]\mathrm{vec}(Y)\\
&\qquad-\sigma^{-4}\mathrm{vec}(\tilde{Y})^*[(\Sigma_G^3\boldsymbol{U}_G^*)\otimes(\Sigma_H\boldsymbol{U}_H^*\nabla_{\theta_i^h}\boldsymbol{H})]\mathrm{vec}(Y)\\
&\qquad-\sigma^{-4}\mathrm{vec}(Y)^*[(\boldsymbol{U}_H\Sigma_H^3)\otimes(\nabla_{\theta_i^h}\boldsymbol{H}\boldsymbol{U}_H\Sigma_H)]\mathrm{vec}(\tilde{Y})\\
&\qquad+\sigma^{-4}\mathrm{vec}(\tilde{Y})^*[\Sigma_G^4\otimes(\Sigma_H\boldsymbol{U}_H^*\nabla_{\theta_i^h}\boldsymbol{H}\boldsymbol{U}_H\Sigma_H))]\mathrm{vec}(\tilde{Y})\\
&\quad=\sigma^{-4}\mathrm{tr}(\Sigma_G^2\boldsymbol{U}_G^\top Y^*\nabla_{\theta_i^h}\boldsymbol{H}Y\overline{\boldsymbol{U}_G})-\sigma^{-4}\mathrm{tr}(\tilde{Y}\Sigma_H\boldsymbol{U}_H^*\nabla_{\theta_i^h}\boldsymbol{H}Y\boldsymbol{U}_G^\top\Sigma_G^3)\\
&\qquad-\sigma^{-4}\mathrm{tr}(\Sigma_G^3\boldsymbol{U}_G^\top Y^*\nabla_{\theta_i^h}\boldsymbol{H}\boldsymbol{U}_H^*\Sigma_H\tilde{Y})+\sigma^{-4}\mathrm{tr}(\Sigma_G^4\tilde{Y}^*\Sigma_H\boldsymbol{U}_H^*\nabla_{\theta_i^h}\boldsymbol{H}\boldsymbol{U}_H\Sigma_H\tilde{Y}^*).
\end{aligned}$$

When $\boldsymbol{H}=\boldsymbol{W}\boldsymbol{W}^*$ and taking derivative with respect to w_{dk},

$$\begin{aligned}
&\mathrm{vec}(Y)^*(\sigma^2\boldsymbol{I}+\boldsymbol{G}\otimes\boldsymbol{H})^{-1}(\boldsymbol{G}\otimes\nabla_{w_{dk}}\boldsymbol{H})(\sigma^2\boldsymbol{I}+\boldsymbol{G}\otimes\boldsymbol{H})^{-1}\mathrm{vec}(Y)\\
&\quad\approx 2\sigma^{-4}(Y_{d:}^\top U_G\Sigma_G^2\boldsymbol{U}_G^*Y^*\boldsymbol{w}_k-Y_{d:}^\top U_G\Sigma_G^3\tilde{Y}^*\Sigma_H\boldsymbol{U}_H^*\boldsymbol{w}_k\\
&\qquad-\boldsymbol{U}_{H,d:}^\top\Sigma_H\tilde{Y}\Sigma_G^3\boldsymbol{U}_G^\top Y^*\boldsymbol{w}_k+\boldsymbol{U}_{H,d:}^\top\Sigma_H\tilde{Y}\Sigma_G^4\tilde{Y}^*\Sigma_H\boldsymbol{U}_H^*\boldsymbol{w}_k).
\end{aligned}$$

These derivative approximations of $\ell(\boldsymbol{\theta}^g,\boldsymbol{\theta}^h)$ imply the effectiveness of our low-rank approximation in terms of computational costs, which are lower than those of the Stegle's method [Stegle et al., 2011, Rakitsch et al., 2013].

3.9 Invariance Under Shuffling Snapshot Pairs

We sometimes know how $(\boldsymbol{y}_{t-1},\boldsymbol{y}_t), t=1,\dots,T$ are paired, but we do not know the correct order of the timepoints. More formally, let $\tau:\{1,\dots,T\}\longrightarrow\{1,\dots,T\}$ be a permutation map. We have the pairs

$$(\boldsymbol{y}_{\tau(t-1)},\boldsymbol{y}_{\tau(t)}),\quad\text{for }\ t=2,\dots,T,$$

as observed data, but the permutation τ is unknown. In other words, the shuffled observation matrices $\tilde{Y}_0, \tilde{Y}_1 \in C^{D\times(T-1)}$ is obtained as

$$\begin{aligned}\tilde{Y}_0 &= (\boldsymbol{y}_{\tau(1)}, \boldsymbol{y}_{\tau(2)}, \dots, \boldsymbol{y}_{\tau(T-1)}),\\ \tilde{Y}_1 &= (\boldsymbol{y}_{\tau(2)}, \boldsymbol{y}_{\tau(3)}, \dots, \boldsymbol{y}_{\tau(T)}),\end{aligned}$$

and there exists a $(T-1)\times(T-1)$ permutation matrix $\boldsymbol{P}$ such that $\tilde{Y}_0 = Y_0\boldsymbol{P}$ and $\tilde{Y}_1 = Y_1\boldsymbol{P}$.

DMD even works in this setting. The solution of (3.7) is $Y_1Y_0^+ =: \bar{\boldsymbol{A}}$ in general. Recall that the pseudo-inverse is given by

$$Y_0^+ = \begin{cases} Y_0^*(Y_0Y_0^*)^{-1} & \text{for } D < T-1 \\ Y_0^{-1} & \text{for } D = T-1 \\ (Y_0^*Y_0)^{-1}Y_0^* & \text{for } D > T-1, \end{cases} \tag{3.22}$$

if $Y_0 \in \mathbb{C}^{D\times(T-1)}$ is full-rank[4]. For every case in (3.22), we can confirm that $\bar{\boldsymbol{A}} = Y_1Y_0^+ = \tilde{Y}_1\tilde{Y}_0^+$ holds. This means that the DMD procedure with the shuffled observation matrices $\tilde{Y}_1$ and $\tilde{Y}_0^+$ produces the same result as the unshuffled observations.

Let us see the invariance of GPKMD under shuffling snapshot pairs. Before marginalizing $\{b_{kl}\}$, GPKMD has the joint likelihood

$$\begin{aligned}&p(Y|\{\boldsymbol{x}_t\}, \{\lambda_k\}, \{\boldsymbol{w}_k\}, \{b_{kl}\}, \sigma^2)\\ &= \prod_{t=1}^{T} \mathcal{CN}\left(\boldsymbol{y}_{\tau(t)} \middle| \sum_{k=1}^{K}\left(\sum_l b_{kl}\psi_l(\boldsymbol{x}_{\tau(t)})\right)\boldsymbol{w}_k, \sigma^2\boldsymbol{I}\right)\\ &\qquad \times \mathcal{CN}\left(\boldsymbol{y}_{\tau(t)} \middle| \sum_{k=1}^{K}\lambda_k\left(\sum_l b_{kl}\psi_l(\boldsymbol{x}_{\tau(t-1)})\right)\boldsymbol{w}_k, \sigma^2\boldsymbol{I}\right),\end{aligned}$$

from (3.11). After the marginalization, the permutation τ effects the GPDM prior as:

$$p(\boldsymbol{x}_1, \dots, \boldsymbol{x}_T) = \mathcal{MN}(X_1\boldsymbol{P}|\boldsymbol{O}, \boldsymbol{I}, \boldsymbol{P}^\top\boldsymbol{K}_X\boldsymbol{P} + s_x^2\boldsymbol{I}).$$

[4] In this subsection we use $t = 1, \dots, T$ for timepoints while $t = 0, \dots, T$ are used in (3.7).

This reformulation preserves the original value because

$$\begin{aligned}
p(\boldsymbol{x}_1, \dots, \boldsymbol{x}_T) &= \mathcal{MN}(\boldsymbol{X}_1\boldsymbol{P}|\boldsymbol{O}, \boldsymbol{I}, \boldsymbol{P}^\top \boldsymbol{K}_X \boldsymbol{P} + s_x^2 \boldsymbol{I}) \\
&\propto \exp\left(-\frac{1}{2}\mathrm{tr}((\boldsymbol{P}^\top \boldsymbol{K}_X \boldsymbol{P} + s_x^2 \boldsymbol{I})^{-1} \boldsymbol{P}^\top \boldsymbol{X}_1^\top \boldsymbol{X}_1 \boldsymbol{P})\right) \\
&\propto \exp\left(-\frac{1}{2}\mathrm{tr}(\boldsymbol{P}\boldsymbol{P}^\top (\boldsymbol{K}_X + s_x^2 \boldsymbol{I})^{-1} \boldsymbol{P}\boldsymbol{P}^\top \boldsymbol{X}_1^\top \boldsymbol{X}_1)\right) \\
&\propto \exp\left(-\frac{1}{2}\mathrm{tr}((\boldsymbol{K}_X + s_x^2 \boldsymbol{I})^{-1} \boldsymbol{X}_1^\top \boldsymbol{X}_1)\right) \propto \mathcal{MN}(\boldsymbol{X}_1|\boldsymbol{O}, \boldsymbol{I}, \boldsymbol{K}_X + s_x^2 \boldsymbol{I})
\end{aligned}$$

holds.

4

Minorization-Maximization for Determinantal Point Processes

4.1 Background

A determinantal point process (DPP) is a probabilistic model that represents the occurrence probability of random subsets of a ground set. Initially, DPPs introduced in statistical mechanics to describe the probabilistic behavior of fermions [Macchi, 1975]. In recent years, broader applications of DPPs have been developed in the machine learning community [Kulesza and Taskar, 2012].

An important feature of DPPs is the presence of negative dependence [Borcea et al., 2009]. There exist some characterizations of negative dependence [Mariet, 2019], and here we consider (pairwise) negative correlation as an example. Letting $\mathcal{A}$ be a random subset, $P(\{i, j\} \subseteq \mathcal{A}) \leq P(i \in \mathcal{A})P(j \in \mathcal{A})$ holds for any pair of items i, j in a ground set when $P(\cdot)$ is defined as a DPP. This means that DPPs can take into account inter-element repulsion, which encourages the occurrence of diverse subsets. This feature aligns with a variety of machine learning applications, such as diversity-promoting image search [Kulesza and Taskar, 2011], recommender systems [Gillenwater et al., 2014], base station configuration for cellular networks [Miyoshi and Shirai, 2014], random design regression [Dereziński et al., 2022], and locating inducing points of sparse variational Gaussian process regression [Burt et al., 2020].

A natural problem on DPPs is efficient learning of the parameters. Since a DPP defined on a finite ground set is parameterized by a positive semidefinite kernel matrix, the learning methods are roughly classified into three approaches: (a) assuming the kernel matrix is full-rank and having no additional structure (full-rank DPPs), (b) assuming the kernel matrix is low-rank (low-rank DPPs), or (c) assuming other tractable structure for the kernel matrix.

So far, some learning methods have been designed for full-rank DPPs. Gillenwater et al. [2014] pioneered the learning problem of DPPs; they developed an EM algorithm for full-rank DPPs. Mariet and Sra [2015] later proposed a fixed-point algorithm for full-rank DPPs. They derived a simple update rule for the kernel matrix and showed its monotonicity by finding its equivalence with a minorization-maximization (MM) algorithm. Their experiments also showed that the fixed-point algorithm is more efficient and stable than the EM algorithm.

Gartrell et al. [2017] introduced low-rank DPPs. Learning low-rank DPPs involves gradient-based optimization. Mariet et al. [2019a] proposed contrastive estimation as an alternative of the maximum likelihood estimation (MLE), while Osogami et al. [2018] incorporated temporal dynamics into low-rank DPPs. A Bayesian extension of low-rank DPPs was also proposed in [Gartrell et al., 2016].

In principle, without special structures, it is difficult to overcome the $O(N^3)$ time complexity for full-rank DPPs and $O(NK^2)$ for low-rank DPPs, where N is the size of the ground set and K is the rank of the kernel matrix. To go beyond these complexities, DPPs with special structure are developed, such as Kronecker DPPs [Mariet and Sra, 2016] and the "diagonal+special low-rank" structure [Dupuy and Bach, 2018].

Our study focuses on learning of full-rank DPPs. While full-rank DPPs are sometimes not suitable for problems with a large ground set, we often want to conduct an exact inference for small- to medium-sized problems. For example, consider a hypothetical application of a DPP. The first step in the data analysis is to assess whether DPP-based modeling is appropriate for our task or not. Even if our final goal is to handle large data, we typically take relatively small data collected provisionally during this assessment phase. In such a situation, we hope to utilize a ready-made learning algorithm: requiring less hyperparameter tuning, easily implementable, well-behaved, and good convergence speed. However, the existing methods for full-rank DPPs have some difficulties; the EM algorithm [Gillenwater et al., 2014] internally requires optimization on a Stiefel manifold, making the learning procedure complicated and unstable. In [Mariet and Sra, 2015], the authors introduced a step size in order to accelerate the fixed-point algorithm, but the step size was fixed throughout the learning.

4.1.1 Contributions

In this chapter, we propose a simple yet powerful learning rule for full-rank DPPs based on the MM algorithm. Our method increases the log-likelihood monotonically and stably, and locally provides a tighter minorizer than the fixed-point algorithm. Our minorizer is concave while the fixed-point algorithm maximizes a non-concave minorizer in the iteration. This means it has no concern about optimization failure in each iteration. Moreover, we also develop an accelerated version of the proposed MM algorithm. Although the accelerated algorithm requires fixed hyperparameters, the step size is determined adaptively in each iteration. We conduct experiments with both synthetic and real-world datasets and our method outperforms the existing methods in most settings.

In summary, our main contributions in this chapter are:

- We present an easy-to-implement learning method for full-rank DPPs based on the MM algorithm. By the property of MM algorithms, our method monotonically increases the log-likelihood.

- We compare the tightness of the minorizers between the existing and proposed methods. The fixed-point algorithm for DPPs proposed in [Mariet and Sra, 2015] can also be viewed as an MM algorithm. Our result indicates that our minorizer locally provides a tighter lower-bound than the existing method. Moreover, our method provides a concave minorizer unlike the exsiting method.

- We derive a generalized form of the minorizer and develop an accelerated algorithm. We also provide an adaptive method to determine the step size values in iterations for the accelerated algorithm.

- We conduct experiments to evaluate learning algorithms for full-rank DPPs using both synthetic and real-world datasets. Our empirical results show the superiority of our method in convergence speed and stability.

4.2 Learning Algorithm

In this chapter, we develop a learning algorithm for $\boldsymbol{L}$-ensembles (2.15). Given M samples denoted by $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_M \subseteq \mathcal{Y}$, our goal is to solve MLE. That is, to find a maximizer of the log-likelihood

$$\begin{aligned} f(\boldsymbol{L}) &= \frac{1}{M} \sum_{m=1}^{M} \log\det([\boldsymbol{L}]_{\mathcal{A}_m}) - \log\det(\boldsymbol{L} + \boldsymbol{I}) \\ &= \frac{1}{M} \sum_{m=1}^{M} \log\det(\boldsymbol{U}_{\mathcal{A}_m} \boldsymbol{L} \boldsymbol{U}_{\mathcal{A}_m}^{\top}) - \log\det(\boldsymbol{L} + \boldsymbol{I}), \end{aligned} \tag{4.1}$$

where $\boldsymbol{U}_{\mathcal{A}_m} \in \{0,1\}^{|\mathcal{A}_m| \times N}$ is the submatrix of $\boldsymbol{I}$ obtained by keeping the rows corresponding to the elements in $\mathcal{A}_m$.

4.2.1 MM Algorithm

A minorization-maximization (MM) algorithm is a powerful meta-algorithm for finding a local maximizer of a generally non-concave objective $f(\theta)$ [Hunter and Lange, 2004, Sun et al., 2017]. The MM algorithm consists of two steps: (i) find a minorizer $g(\theta|\theta^{(t)})$ of $f(\theta)$ that satisfies

- $f(\theta) \geq g(\theta|\theta^{(t)})$
- $f(\theta^{(t)}) = g(\theta^{(t)}|\theta^{(t)})$

for all θ and $\theta^{(t)}$ within a feasible region. Then, (ii) maximize the minorizer $g(\theta|\theta^{(t)})$ with respect to θ and set $\theta^{(t+1)} = \arg\max g(\theta|\theta^{(t)})$. Repeating this process, we can obtain a sequence of the parameters $\{\theta^{(t)}\}_{t \geq 0}$ which monotonically increases the objective value, because

$$f(\theta^{(t+1)}) \geq g(\theta^{(t+1)}|\theta^{(t)}) \geq g(\theta^{(t)}|\theta^{(t)}) = f(\theta^{(t)}) \tag{4.2}$$

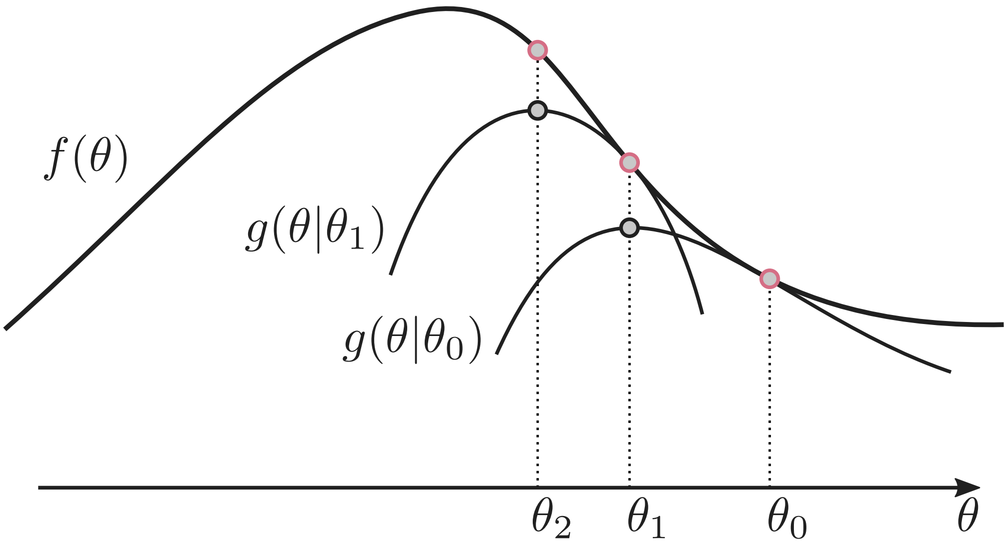


Figure 4.1: Outline of an MM algorithm. The yielded parameters (shown in pink) increase the objective function $f(\theta)$ monotonically.

holds. Figure 4.1 outlines the MM algorithm. We can see that the yielded parameters $\theta_0, \theta_1, \theta_2, \ldots$ increase the objective function $f(\theta)$ monotonically.

Since $\log\det(\cdot)$ is concave on $\mathbb{S}_{++}$, the objective function (4.1) is a combination of concave and convex functions. From the concavity of $\log\det(\cdot)$, the following linear upper bound is derived with the first-order Taylor expansion

$$\log\det(\boldsymbol{X}) \leq \log\det(\boldsymbol{Y}) + \mathrm{tr}\{\boldsymbol{Y}^{-1}(\boldsymbol{X} - \boldsymbol{Y})\} = \log\det(\boldsymbol{Y}) + \mathrm{tr}(\boldsymbol{Y}^{-1}\boldsymbol{X}) - n \tag{4.3}$$

for any $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{S}^n_{++}, n \in \mathbb{N}$, and by swapping $\boldsymbol{X}$ and $\boldsymbol{Y}$,

$$\log\det(\boldsymbol{X}) \geq \log\det(\boldsymbol{Y}) - \mathrm{tr}\{\boldsymbol{X}^{-1}(\boldsymbol{Y} - \boldsymbol{X})\} = \log\det(\boldsymbol{Y}) - \mathrm{tr}(\boldsymbol{X}^{-1}\boldsymbol{Y}) + n \tag{4.4}$$

also holds.

From (4.3) with $\boldsymbol{X} \to \boldsymbol{L} + \boldsymbol{I}$ and $\boldsymbol{Y} \to \boldsymbol{L}^{(t+1)} + \boldsymbol{I}$, we have

$$-\log\det(\boldsymbol{L} + \boldsymbol{I}) \geq -\log\det(\boldsymbol{L}^{(t)} + \boldsymbol{I}) - \mathrm{tr}\{(\boldsymbol{L}^{(t)} + \boldsymbol{I})^{-1}(\boldsymbol{L} - \boldsymbol{L}^{(t)})\}, \tag{4.5}$$

which yields a choice for minorizing the objective (4.1). This method is referred to as the concave-convex procedure (CCCP) [Yuille and Rangarajan, 2001], a special case of

MM algorithms. However, the minorizer derived by the CCCP has no closed-form maximizer in our case, therefore, we devise an easy-to-optimize alternative.

4.2.2 Proposed Algorithm

In the proposed minorizer of (4.1), the convex part is lower-bounded linearly by (4.5) and the concave part $\log\det([\boldsymbol{L}]_{\mathcal{A}_m}) = \log\det(\boldsymbol{U}_{\mathcal{A}_m}\boldsymbol{L}\boldsymbol{U}_{\mathcal{A}_m}^\top)$ is also lower-bounded. The following proposition provides the concrete form of our proposed minorizer.

Proposition 4.1. *Let $f(\boldsymbol{L})$ be given by (4.1) and*

$$\begin{aligned} &g(\boldsymbol{L}|\boldsymbol{L}^{(t)}) \\ &\quad = -\frac{1}{M}\sum_{m=1}^{M}\mathrm{tr}\{\boldsymbol{L}^{(t)}\boldsymbol{U}_{\mathcal{A}_m}^\top[\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}^{-1}\boldsymbol{U}_{\mathcal{A}_m}\boldsymbol{L}^{(t)}\boldsymbol{L}^{-1}\} - \mathrm{tr}\{(\boldsymbol{L}^{(t)}+\boldsymbol{I})^{-1}\boldsymbol{L}\} + \zeta(\boldsymbol{L}^{(t)}), \end{aligned} \tag{4.6}$$

where

$$\begin{aligned} &\zeta(\boldsymbol{L}^{(t)}) \\ &= \frac{1}{M}\sum_{m=1}^{M}\left\{\log\det(\boldsymbol{U}_{\mathcal{A}_m}\boldsymbol{L}^{(t)}\boldsymbol{U}_{\mathcal{A}_m}^\top) + |\mathcal{A}_m|\right\} - \log\det(\boldsymbol{L}^{(t)}+\boldsymbol{I}) + \mathrm{tr}\{(\boldsymbol{L}^{(t)}+\boldsymbol{I})^{-1}\boldsymbol{L}^{(t)}\} \end{aligned}$$

is a constant term. Then, $f(\boldsymbol{L}) \geq g(\boldsymbol{L}|\boldsymbol{L}^{(t)})$ and $f(\boldsymbol{L}^{(t)}) = g(\boldsymbol{L}^{(t)}|\boldsymbol{L}^{(t)})$ hold for any $\boldsymbol{L}, \boldsymbol{L}^{(t)} \in \mathbb{S}_{++}^N$.

Proof. For any positive definite $\boldsymbol{P}, \boldsymbol{P}_t \succ 0$ and any square or broad non-degenerate matrix $\boldsymbol{A}$, the following matrix inequality holds [Sun et al., 2016, 2017]:

$$\begin{aligned} (\boldsymbol{A}\boldsymbol{P}\boldsymbol{A}^\top)^{-1} &\preceq \boldsymbol{R}_t^{-1}\boldsymbol{A}\boldsymbol{P}_t\boldsymbol{P}^{-1}\boldsymbol{P}_t\boldsymbol{A}^\top\boldsymbol{R}_t^{-1}, \\ \boldsymbol{R}_t &= \boldsymbol{A}\boldsymbol{P}_t\boldsymbol{A}^\top, \end{aligned}$$

and thus we have

$$\mathrm{tr}\{(\boldsymbol{A}\boldsymbol{P}\boldsymbol{A}^\top)^{-1}\boldsymbol{S}\} \leq \mathrm{tr}\{\boldsymbol{R}_t^{-1}\boldsymbol{A}\boldsymbol{P}_t\boldsymbol{P}^{-1}\boldsymbol{P}_t\boldsymbol{A}^\top\boldsymbol{R}_t^{-1}\boldsymbol{S}\} \tag{4.7}$$

for any appropriately sized and positive semidefinite $\boldsymbol{S} \succeq \boldsymbol{O}$. Using the lower-bound (4.4) with the substitutions $\boldsymbol{X} \to \boldsymbol{U}_{\mathcal{A}_m}\boldsymbol{L}\boldsymbol{U}_{\mathcal{A}_m}^\top$, $\boldsymbol{Y} \to \boldsymbol{U}_{\mathcal{A}_m}\boldsymbol{L}^{(t)}\boldsymbol{U}_{\mathcal{A}_m}^\top$ and (4.7) with $\boldsymbol{A} \to \boldsymbol{U}_{\mathcal{A}_m}, \boldsymbol{P} \to \boldsymbol{L}, \boldsymbol{P}_t \to \boldsymbol{L}^{(t)}$ and $\boldsymbol{S} \to \boldsymbol{U}_{\mathcal{A}_m}\boldsymbol{L}^{(t)}\boldsymbol{U}_{\mathcal{A}_m}^\top$, we have

$$\begin{aligned}\log\det(\boldsymbol{U}_{\mathcal{A}_m}\boldsymbol{L}\boldsymbol{U}_{\mathcal{A}_m}^\top) &\geq |\mathcal{A}_m| + \log\det(\boldsymbol{U}_{\mathcal{A}_m}\boldsymbol{L}^{(t)}\boldsymbol{U}_{\mathcal{A}_m}^\top) - \operatorname{tr}\{(\boldsymbol{U}_{\mathcal{A}_m}\boldsymbol{L}\boldsymbol{U}_{\mathcal{A}_m}^\top)^{-1}\boldsymbol{U}_{\mathcal{A}_m}\boldsymbol{L}^{(t)}\boldsymbol{U}_{\mathcal{A}_m}^\top\} \\ &\geq |\mathcal{A}_m| + \log\det(\boldsymbol{U}_{\mathcal{A}_m}\boldsymbol{L}^{(t)}\boldsymbol{U}_{\mathcal{A}_m}^\top) - \operatorname{tr}\{\boldsymbol{L}^{(t)}\boldsymbol{U}_{\mathcal{A}_m}^\top[\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}^{-1}\boldsymbol{U}_{\mathcal{A}_m}\boldsymbol{L}^{(t)}\boldsymbol{L}^{-1}\}.\end{aligned} \tag{4.8}$$

Combining the lower-bounds (4.5) and (4.8), we can construct the minorizer of $f(\boldsymbol{L})$ as (4.6). □

In order to obtain the maximizer of (4.1), we iteratively optimize the proposed minorizer $g(\boldsymbol{L}|\boldsymbol{L}^{(t)})$ by solving the first-order optimality condition for $t = 1, \ldots, T$. Since $g(\boldsymbol{L}|\boldsymbol{L}^{(t)})$ is concave because of the convexity of $\operatorname{tr}(\boldsymbol{X}^{-1})$ for $\boldsymbol{X} \succ 0$, a stationary point of $g(\boldsymbol{L}|\boldsymbol{L}^{(t)})$ is also its global maximizer.

Proposition 4.2. *A global maximizer of $g(\boldsymbol{L}|\boldsymbol{L}^{(t)})$ satisfies*

$$-\boldsymbol{L}(\boldsymbol{L}^{(t)} + \boldsymbol{I})^{-1}\boldsymbol{L} + \boldsymbol{Q}_M^{(t)} = \boldsymbol{O}, \tag{4.9}$$

where

$$\boldsymbol{Q}_M^{(t)} = \boldsymbol{L}^{(t)}\left(\frac{1}{M}\sum_{m=1}^{M}\boldsymbol{U}_{\mathcal{A}_m}^\top[\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}^{-1}\boldsymbol{U}_{\mathcal{A}_m}\right)\boldsymbol{L}^{(t)}. \tag{4.10}$$

Proof. Noting that $\nabla_{\boldsymbol{X}}\operatorname{tr}(\boldsymbol{A}\boldsymbol{X}) = \boldsymbol{A}^\top$ and $\nabla_{\boldsymbol{X}}\operatorname{tr}(\boldsymbol{A}\boldsymbol{X}^{-1}) = -(\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}^{-1})^\top$ for appropriate matrices $\boldsymbol{X}$ and $\boldsymbol{A}$, the optimality condition of (4.6) is

$$\nabla_{\boldsymbol{L}}g(\boldsymbol{L}|\boldsymbol{L}^{(t)}) = \boldsymbol{L}^{-1}\boldsymbol{Q}_M^{(t)}\boldsymbol{L}^{-1} - (\boldsymbol{L}^{(t)} + \boldsymbol{I})^{-1} = \boldsymbol{O}. \tag{4.11}$$

By multiplying both sides of (4.11) by $\boldsymbol{L}$, we can see that the stationary points of $g(\boldsymbol{L}|\boldsymbol{L}^{(t)})$ satisfy (4.9). From the concavity of $g(\boldsymbol{L}|\boldsymbol{L}^{(t)})$, we obtain the result. □

The matrix quadratic equation (4.9) is a special case of the continuous algebraic

Algorithm 1: Minorization-Maximization (MM)

```
Input: Training set $\{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_M\}$, initial value $\boldsymbol{L} \succ \boldsymbol{O}$, and machine epsilon
       $\varepsilon \geq 0$
Output: $\boldsymbol{L}$
for $t = 1$ to $T$ do
    $\boldsymbol{A} \leftarrow \boldsymbol{O}$;
    $\boldsymbol{Q}_\varepsilon \leftarrow \boldsymbol{L}\left(\frac{1}{M}\sum_{m=1}^{M} \boldsymbol{U}_{\mathcal{A}_m}^\top [\boldsymbol{L}]_{\mathcal{A}_m}^{-1} \boldsymbol{U}_{\mathcal{A}_m}\right)\boldsymbol{L} + \varepsilon \boldsymbol{I}$;
    $\boldsymbol{G} \leftarrow (\boldsymbol{L} + \boldsymbol{I})^{-1}$;
    $\boldsymbol{L} \leftarrow \mathrm{SolveCARE}(\boldsymbol{A}, \boldsymbol{Q}_\varepsilon, \boldsymbol{G})$; // Solve Equation (4.12)
end
```

Riccati equation (CARE):

$$\boldsymbol{A}^\top \boldsymbol{X} + \boldsymbol{X}\boldsymbol{A} - \boldsymbol{X}\boldsymbol{G}\boldsymbol{X} + \boldsymbol{Q} = \boldsymbol{O}, \tag{4.12}$$

where $\boldsymbol{X} \in \mathbb{S}^N$ is unknown, and $\boldsymbol{G}, \boldsymbol{Q} \in \mathbb{S}^N$, $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ are fixed coefficient matrices. CARE is well-studied in control engineering and is solvable by some numerical methods such as the Schur method [Laub, 1979] and Newton's method [Bini et al., 2011, Benner and Byers, 1998]. It is worth noting that CARE solvers are available in most programming languages through packages for scientific computation; for example, `SciPy` in Python and `MatrixEquations.jl` in Julia.

In addition, we can confirm the following statement as a corollary of Proposition 4.2.

Corollary 4.1. *With the same notation as in Proposition 4.2 and a positive definite initial value* $\boldsymbol{L}^{(0)} \succ 0$, *we have* $\mathrm{rank}(\boldsymbol{L}^{(t)}) = \mathrm{rank}(\boldsymbol{Q}_M^{(t)})$ *for* $t = 1, 2, \ldots$.

Proof. Since $\boldsymbol{L}^{(0)}$ is positive definite, $(\boldsymbol{L}^{(0)} + \boldsymbol{I})^{-1}$ is non-singular. Therefore, from the optimality condition (4.9),

$$\mathrm{rank}(\boldsymbol{L}^{(1)}(\boldsymbol{L}^{(0)} + \boldsymbol{I})^{-1}\boldsymbol{L}^{(1)}) = \mathrm{rank}(\boldsymbol{L}^{(1)}) = \mathrm{rank}(\boldsymbol{Q}_M^{(1)}).$$

By applying similar operations recursively, the result can be confirmed. □

From the assumption in Corollary 4.1, we find that $\boldsymbol{L}^{(0)}$ should be initialized by some positive definite matrix. See the experimental settings described in Section 4.4

for examples of the initialization. Corollary 4.1 also says that if $Q_M^{(t)}$ is degenerate, the solution of (4.9) must also be degenerate. This means that when $Q_M^{(t)}$ is singular, the solution of (4.9) falls outside the feasible region $\mathbb{S}_{++}^N$. This problem arises when some elements of $\mathcal{Y}$ are never observed in the given data $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_M$, that is, $\bigcup_{m=1}^M \mathcal{A}_m \subsetneq \mathcal{Y}$ holds. To avoid this issue and stabilize the numerical computation, we recommend to solve

$$-\boldsymbol{L}(\boldsymbol{L}^{(t)} + \boldsymbol{I})^{-1}\boldsymbol{L} + \boldsymbol{Q}_M^{(t)} + \varepsilon \boldsymbol{I} = \boldsymbol{O}$$

with a machine epsilon $\varepsilon > 0$ instead of (4.9). We note that the choice of the machine epsilon ε does not affect the estimate significantly; we use $\varepsilon = 10^{-10}$ throughout this chapter. The procedure for the proposed MM-based learning is summarized in Algorithm 1.

4.2.3 Relation to the Existing Method

Mariet and Sra [2015] derived the following update rule to maximize (4.1) as a fixed-point algorithm:

$$\boldsymbol{L}^{(t+1)} = \boldsymbol{L}^{(t)} + a\boldsymbol{L}^{(t)}\nabla f(\boldsymbol{L}^{(t)})\boldsymbol{L}^{(t)}, \tag{4.13}$$

$$\nabla f(\boldsymbol{L}) = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{U}_{\mathcal{A}_m}^\top [\boldsymbol{L}]_{\mathcal{A}_m}^{-1} \boldsymbol{U}_{\mathcal{A}_m} - (\boldsymbol{L} + \boldsymbol{I})^{-1},$$

where $a > 0$ is a step size. For $a = 1$, they also show that the update rule (4.13) can also be regarded as an MM algorithm with the non-concave minorizer

$$\begin{aligned} h(\boldsymbol{L}|\boldsymbol{L}^{(t)}) = &-\frac{1}{M}\sum_{m=1}^{M} \operatorname{tr}\{\boldsymbol{L}^{(t)}\boldsymbol{U}_{\mathcal{A}_m}^\top [\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}^{-1} \boldsymbol{U}_{\mathcal{A}_m} \boldsymbol{L}^{(t)} \boldsymbol{L}^{-1}\} \\ &- \log\det(\boldsymbol{L}) - \operatorname{tr}\{(\boldsymbol{L}^{(t)} + \boldsymbol{I})^{-1}\boldsymbol{L}^{-1}\boldsymbol{L}^{(t)}\} + \xi(\boldsymbol{L}^{(t)}), \end{aligned} \tag{4.14}$$

where $\xi(\boldsymbol{L}^{(t)})$ is a constant term and explicitly given in Section 4.6. Comparing (4.14) with (4.6), we can see that the lower-bounds for the first term in (4.1) are the same, and those for the second term only differ. With respect to these minorizers, the following proposition holds.

Proposition 4.3. *For $g(\boldsymbol{L}|\boldsymbol{L}^{(t)})$ defined in* (4.6) *and $h(\boldsymbol{L}|\boldsymbol{L}^{(t)})$ defined in* (4.14)*, it holds that $g(\boldsymbol{L}|\boldsymbol{L}^{(t)}) \geq h(\boldsymbol{L}|\boldsymbol{L}^{(t)})$ for* **L** *in the δ-neighborhood of $\boldsymbol{L}^{(t)}$: $\mathcal{B}_\delta(\boldsymbol{L}^{(t)}) = \{\boldsymbol{L}^{(t)} + \delta \boldsymbol{M}$: $\boldsymbol{M}$ is a symmetric matrix whose eigenvalues are all in $[-1, 1]\}$ with a sufficiently small $\delta > 0$.*

Proof. We have the following inequality:

$$g(\boldsymbol{L}|\boldsymbol{L}^{(t)}) - h(\boldsymbol{L}|\boldsymbol{L}^{(t)}) \geq \mathrm{tr}\{(\boldsymbol{L}^{(t)} + \boldsymbol{I})^{-1}(2\boldsymbol{L}^{(t)} - \boldsymbol{L} - \boldsymbol{L}^{(t)}\boldsymbol{L}^{-1}\boldsymbol{L}^{(t)})\}, \tag{4.15}$$

where the derivation is shown in Section 4.6. If $\boldsymbol{L} \in \mathcal{B}_\delta(\boldsymbol{L}^{(t)})$, we have

$$2\boldsymbol{L}^{(t)} - \boldsymbol{L} - \boldsymbol{L}^{(t)}\boldsymbol{L}^{-1}\boldsymbol{L}^{(t)} \approx \boldsymbol{O}. \tag{4.16}$$

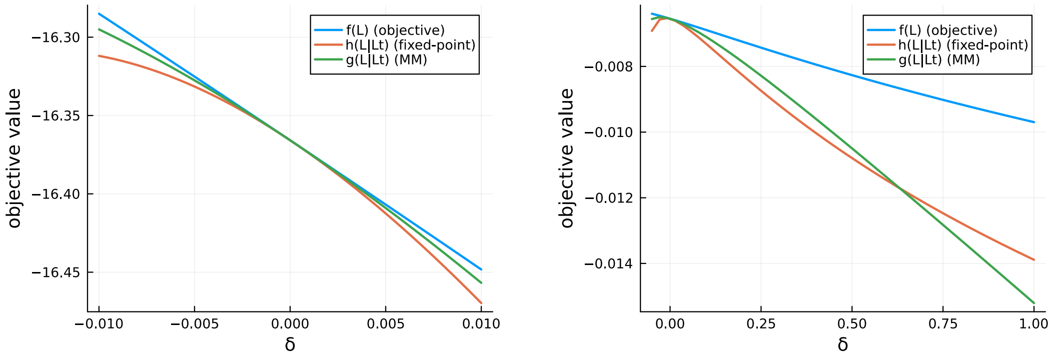
Details of the derivation can be found in Section 4.6. Applying the approximation (4.16) to (4.15), we can conclude the proposition. □

Proposition 4.3 states that the proposed minorizer gives a tighter lower-bound of the objective than that of the existing method locally. This leads to a tighter leftmost inequality in (4.2), making it likely that the proposed method will produce better $\boldsymbol{L}^{(t+1)}$. Figure 4.2 shows the behavior of the minorizers in the neighborhood and non-neighborhood of $\boldsymbol{L}^{(t)}$. The proposed minorizer becomes looser as $\boldsymbol{L}$ moves farther away from $\boldsymbol{L}^{(t)}$, but the experimental results in Section 4.4 show that the proposed method converges faster in most cases. Note that the minorizer of the fixed-point algorithm is non-convex as seen in Figure 4.2b. This implies that the fixed-point algorithm is possible to get trapped in poor stationary points of $h(\boldsymbol{L}|\boldsymbol{L}^{(t)})$.

4.2.4 Computational Costs

In our method, the total computational cost per iteration is $O(M\kappa^3 + N^3)$, where $\kappa = \max_m |\mathcal{A}_m|$. It is computed as follows; the computation of $\boldsymbol{Q}_M^{(t)}$ in (4.9) requires $O(\sum_{m=1}^M |\mathcal{A}_m|^3 + N^3) = O(M\kappa^3 + N^3)$ operations, including the evaluation of $[\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}^{-1}$ for all $m = 1, 2, \ldots, M$ and the matrix multiplications of the $N \times N$ matrices. The inversion $(\boldsymbol{L}^{(t)} + \boldsymbol{I})^{-1}$ and solving the CARE also cost $O(N^3)$.

The computational complexity of our method is equal to that of the fixed-point algorithm [Mariet and Sra, 2015]. Although our method incurs additional $O(N^3)$



(a) Neighborhood of $\boldsymbol{L}^{(t)}$. (b) Non-neighborhood of $\boldsymbol{L}^{(t)}$.

Figure 4.2: Behavior of minorizers.

computations due to the CARE, the experimental results in Section 4.4 show faster convergence of our method in computational time. We note that gradient-based learning of a low-rank factorized DPP also takes the same $O(M\kappa^3 + N^3)$ per iteration if the factorization is full-rank [Gartrell et al., 2017, Osogami et al., 2018].

4.3 Generalization and Acceleration

In this section, we develop generalization of the minorizer (4.6) and the CARE (4.9) for further acceleration of the algorithm.

4.3.1 Generalizing the Minorizer

By adding a penalty term to the mean log-likelihood (4.1), we can generalize the objective as

$$f_{\mu^{(t)}}(\boldsymbol{L}|\boldsymbol{L}^{(t)}) = f(\boldsymbol{L}) - \mu^{(t)} d(\boldsymbol{L}\|\boldsymbol{L}^{(t)}), \tag{4.17}$$

where $\mu^{(t)} \geq 0$ is a non-negative coefficient and $d(\cdot\|\cdot)$ is an appropriate divergence defined on $\mathbb{S}^N_{++} \times \mathbb{S}^N_{++}$. The additional penalty term $\mu^{(t)} d(\boldsymbol{L}\|\boldsymbol{L}^{(t)})$ effects to prevent a big change from $\boldsymbol{L}^{(t)}$ to $\boldsymbol{L}^{(t+1)}$. By the definition of a divergence, $d(\boldsymbol{L}\|\boldsymbol{L}^{(t)}) \geq 0$ for any $\boldsymbol{L}, \boldsymbol{L}^{(t)} \in \mathbb{S}^N_{++}$ and the equality holds if and only if $\boldsymbol{L} = \boldsymbol{L}^{(t)}$. This means that the

Algorithm 2: Accelerated MM

```
Input: Training set $\{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_M\}$, initial value $\boldsymbol{L} \succ \boldsymbol{O}$, machine epsilon
       $\varepsilon \geq 0$, tolerance $\delta \in (0, 1)$, and acceleration steps $T_{\text{acc}} \in \{0, 1, \ldots, T\}$
Output: $\boldsymbol{L}$
for $t = 1$ to $T$ do
    $\boldsymbol{H} \leftarrow \left( \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{U}_{\mathcal{A}_m}^{\top} [\boldsymbol{L}]_{\mathcal{A}_m}^{-1} \boldsymbol{U}_{\mathcal{A}_m} \right)$;
    if $t \leq T_{\text{acc}}$ then
        $\mu \leftarrow \min \{\max \{-1/\lambda_{\max} (\boldsymbol{H}(\boldsymbol{L} + \boldsymbol{I})), -1\} + \delta, 0\}$;
    end
    else
        $\mu \leftarrow 0$;
    end
    $\boldsymbol{A} \leftarrow \boldsymbol{O}$;
    $\boldsymbol{Q}_{\mu,\varepsilon} \leftarrow (1 + \mu)\boldsymbol{LHL} + \varepsilon \boldsymbol{I}$;
    $\boldsymbol{G}_{\mu} \leftarrow \mu \boldsymbol{H} + (\boldsymbol{L} + \boldsymbol{I})^{-1}$;
    $\boldsymbol{L} \leftarrow \text{SolveCARE}(\boldsymbol{A}, \boldsymbol{Q}_{\mu,\varepsilon}, \boldsymbol{G}_{\mu})$; // Solve Equation (4.12)
end
```

generalized objective $f_{\mu^{(t)}}(\boldsymbol{L}|\boldsymbol{L}^{(t)})$ also works as the minorizer of $f(\boldsymbol{L})$. Specifically, such a scheme is called the proximal point algorithm if the divergence $d(\cdot\|\cdot)$ is the squared Euclidean distance [Parikh and Boyd, 2014]. Or it is also called mirror ascent (descent) or Bregman minorization (majorization) if $d(\cdot\|\cdot)$ is a Bregman divergence [Nemirovsky, 1983, Beck and Teboulle, 2003, Lange et al., 2021].

In our case, we consider a logdet divergence:

$$D_{\text{ld}}(\boldsymbol{X}\|\boldsymbol{Y}) = -\log\det(\boldsymbol{X}) + \log\det(\boldsymbol{Y}) + \operatorname{tr}\{\boldsymbol{Y}^{-1}(\boldsymbol{X} - \boldsymbol{Y})\},$$

and define $d(\cdot\|\cdot)$ as

$$d(\boldsymbol{L}\|\boldsymbol{L}^{(t)}) = \frac{1}{M} \sum_{m=1}^{M} D_{\text{ld}}([\boldsymbol{L}^{(t)}]_{\mathcal{A}_m} \| [\boldsymbol{L}]_{\mathcal{A}_m}). \tag{4.18}$$

The defined $d(\cdot\|\cdot)$ in (4.18) satisfies the definition of a divergence if and only if $\bigcup_m \mathcal{A}_m = \mathcal{Y}$ holds. The divergence (4.18) leads the following minorizer of $f(\boldsymbol{L})$ and $f_{\mu^{(t)}}(\boldsymbol{L}|\boldsymbol{L}^{(t)})$.

Proposition 4.4. *Let $f(\boldsymbol{L})$ be defined in (4.1) and $f_{\mu^{(t)}}(\boldsymbol{L}|\boldsymbol{L}^{(t)})$ be defined in (4.17) with $\mu^{(t)} \geq 0$ and the divergence (4.18). Then, the concave function*

$$g_{\mu^{(t)}}(\boldsymbol{L}|\boldsymbol{L}^{(t)}) = -\frac{1+\mu^{(t)}}{M}\sum_{m=1}^{M}\mathrm{tr}(\boldsymbol{L}^{(t)}\boldsymbol{U}_{\mathcal{A}_m}^{\top}[\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}^{-1}\boldsymbol{U}_{\mathcal{A}_m}\boldsymbol{L}^{(t)}\boldsymbol{L}^{-1}) \\ -\frac{\mu^{(t)}}{M}\sum_{m=1}^{M}\mathrm{tr}([\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}^{-1}[\boldsymbol{L}]_{\mathcal{A}_m}) - \mathrm{tr}\{(\boldsymbol{L}^{(t)}+\boldsymbol{I})^{-1}\boldsymbol{L}\} + \zeta_{\mu^{(t)}}(\boldsymbol{L}^{(t)}),$$

where $\zeta_{\mu^{(t)}}(\boldsymbol{L}^{(t)})$ is a constant term, is the minorizer of $f(\boldsymbol{L})$ and $f_{\mu^{(t)}}(\boldsymbol{L}|\boldsymbol{L}^{(t)})$.

See Section 4.7 for the proof.

We can maximize $g_{\mu^{(t)}}(\cdot|\boldsymbol{L}^{(t)})$ by solving a CARE in the same manner as Proposition 4.2.

Proposition 4.5. *A global maximizer of $g_{\mu^{(t)}}(\boldsymbol{L}|\boldsymbol{L}^{(t)})$ satisfies the CARE*

$$-\boldsymbol{L}\left\{\mu^{(t)}\boldsymbol{H}_M^{(t)} + (\boldsymbol{L}^{(t)}+\boldsymbol{I})^{-1}\right\}\boldsymbol{L} + (1+\mu^{(t)})\boldsymbol{L}^{(t)}\boldsymbol{H}_M^{(t)}\boldsymbol{L}^{(t)} = \boldsymbol{O}, \tag{4.19}$$

where

$$\boldsymbol{H}_M^{(t)} = \frac{1}{M}\sum_{m=1}^{M}\boldsymbol{U}_{\mathcal{A}_m}^{\top}[\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}^{-1}\boldsymbol{U}_{\mathcal{A}_m}.$$

$\boldsymbol{H}_M^{(t)}$ degenerates if $\bigcup_{m=1}^{M}\mathcal{A}_m \subsetneq \mathcal{Y}$ holds as well as $\boldsymbol{Q}_M^{(t)} = \boldsymbol{L}^{(t)}\boldsymbol{H}_M^{(t)}\boldsymbol{L}^{(t)}$ defined in (4.10). For $\mu^{(t)} = 0$, we have $g_{\mu^{(t)}}(\boldsymbol{L}|\boldsymbol{L}^{(t)}) = g(\boldsymbol{L}|\boldsymbol{L}^{(t)})$ and the update rule (4.19) comes down to the original CARE (4.9). For $\mu^{(t)} > 0$, the update rule (4.19) also works as the MM iteration but the convergence may become slower by the penalty term.

4.3.2 Acceleration and Hyperparameter Determination

What happens if the coefficient $\mu^{(t)}$ is set to negative? Then, $f_{\mu^{(t)}}(\boldsymbol{L}|\boldsymbol{L}^{(t)})$ and $g_{\mu^{(t)}}(\boldsymbol{L}|\boldsymbol{L}^{(t)})$ can no longer be regarded as the minorizers, but it is expected that the update rule produces a bigger change from $\boldsymbol{L}^{(t)}$ to $\boldsymbol{L}^{(t+1)}$ and the learning speed may become faster. However, similar to the learning rate of a gradient descent, a too large absolute value for $\mu^{(t)} < 0$ may lead to bad convergence. Worse still, the solution of the CARE (4.19) can even not exist. Our approach to decide the negative $\mu^{(t)} < 0$ is to ensure that there is at least a feasible solution to the CARE (4.19).

Lemma 4.1. *Let $\boldsymbol{G} \in \mathbb{S}^N$ and $\boldsymbol{Q} \in \mathbb{S}^N_{++}$ be fixed coefficients and $\boldsymbol{X} \in \mathbb{S}^N$ be unknown. Then, the following equation*

$$\boldsymbol{X}\boldsymbol{G}\boldsymbol{X} = \boldsymbol{Q} \tag{4.20}$$

has a solution in $\mathbb{S}^N_{++}$ if and only if $\boldsymbol{G} \succ \boldsymbol{O}$.

Proof. If $\boldsymbol{G}$ is not positive definite, any $\boldsymbol{X}$ does not satisfy (4.20). Taking the contrapositive, if $\boldsymbol{X}$ is the solution of (4.20), $\boldsymbol{G}$ must be positive definite. Conversely, if $\boldsymbol{G}$ is positive definite, $\boldsymbol{G}^{\frac{1}{2}} \succ \boldsymbol{O}$ exists and the equation (4.20) becomes $\boldsymbol{X}\boldsymbol{G}^{\frac{1}{2}}\boldsymbol{G}^{\frac{1}{2}}\boldsymbol{X} = \boldsymbol{Q}^{\frac{1}{2}}\boldsymbol{Q}^{\frac{1}{2}}$. We thus have $\boldsymbol{X}\boldsymbol{G}^{\frac{1}{2}} = \boldsymbol{Q}^{\frac{1}{2}}$ and the equation has the solution $\boldsymbol{X} = \boldsymbol{Q}^{\frac{1}{2}}\boldsymbol{G}^{-\frac{1}{2}} \in \mathbb{S}^N_{++}$. □

Proposition 4.6. *Suppose $\boldsymbol{H}_M^{(t)} \succ \boldsymbol{O}$. Then, the CARE (4.19) has a solution in $\mathbb{S}^N_{++}$ if*

$$\mu^{(t)} > \max\{-1, -1/\lambda_{\max}(\boldsymbol{H}_M^{(t)}(\boldsymbol{L}^{(t)} + \boldsymbol{I}))\}, \tag{4.21}$$

where $\lambda_{\max}(\boldsymbol{X})$ denotes the largest eigenvalue of $\boldsymbol{X}$.

Proof. The right-hand side of the following CARE

$$\boldsymbol{L}\left\{\mu^{(t)}\boldsymbol{H}_M^{(t)} + (\boldsymbol{L}^{(t)} + \boldsymbol{I})^{-1}\right\}\boldsymbol{L} = (1 + \mu^{(t)})\boldsymbol{L}^{(t)}\boldsymbol{H}_M^{(t)}\boldsymbol{L}^{(t)} \tag{4.22}$$

is positive definite by the conditions. For $\mu^{(t)} \geq 0$, the solution of (4.22) immediately exists by Lemma 4.1. When $-1 < \mu^{(t)} < 0$, we can see that the solution exists if and only if

$$\mu^{(t)}\boldsymbol{H}_M^{(t)} + (\boldsymbol{L}^{(t)} + \boldsymbol{I})^{-1} \succ \boldsymbol{O}$$

also from Lemma 4.1. Then, we have

$$
\begin{aligned}
\mu^{(t)} \boldsymbol{H}_M^{(t)} + (\boldsymbol{L}^{(t)} + \boldsymbol{I})^{-1} \succ \boldsymbol{O} \iff{}& \boldsymbol{H}_M^{(t)\frac{1}{2}} (\boldsymbol{I} + \mu^{(t)-1} \boldsymbol{H}_M^{(t)-\frac{1}{2}} (\boldsymbol{L}^{(t)} + \boldsymbol{I})^{-1} \boldsymbol{H}_M^{(t)-\frac{1}{2}}) \boldsymbol{H}_M^{(t)\frac{1}{2}} \prec \boldsymbol{O} \\
\iff{}& \boldsymbol{I} + \mu^{(t)-1} \boldsymbol{H}_M^{(t)-\frac{1}{2}} (\boldsymbol{L}^{(t)} + \boldsymbol{I})^{-1} \boldsymbol{H}_M^{(t)-\frac{1}{2}} \prec \boldsymbol{O} \\
\iff{}& \mu^{(t)} \boldsymbol{I} \succ -\boldsymbol{H}_M^{(t)-\frac{1}{2}} (\boldsymbol{L}^{(t)} + \boldsymbol{I})^{-1} \boldsymbol{H}_M^{(t)-\frac{1}{2}} \\
\iff{}& \mu^{(t)} > \lambda_{\max}(-\boldsymbol{H}_M^{(t)-1} (\boldsymbol{L}^{(t)} + \boldsymbol{I})^{-1}) \\
\iff{}& \mu^{(t)} > -1/\lambda_{\max}(\boldsymbol{H}_M^{(t)} (\boldsymbol{L}^{(t)} + \boldsymbol{I})).
\end{aligned}
$$

□

In the accelerated algorithm, the inequality (4.21) should be satisfied strictly. Algorithm 2 shows the entire procedure of our MM-based learning with acceleration on the basis of Proposition 4.6. In Algorithm 2, we introduce two hyperparameters; one is a tolerance $\delta > 0$ that guarantees the inequality (4.21) strictly and $T_{\text{acc}} \in \{0, 1, \dots, T\}$ denotes up to how many iterations the acceleration is applied. We can automatically adjust the step size coefficient $\mu^{(t)}$ at each iteration within the algorithm with fixed $\delta > 0$, while user-defined fixed step size coefficients are used in the existing fixed-point algorithm [Mariet and Sra, 2015]. In the resulting algorithm, we decide the step size by

$$
\mu = \min\left\{\max\left\{-1/\lambda_{\max}\left(\boldsymbol{H}_M^{(t)}(\boldsymbol{L}^{(t)} + \boldsymbol{I})\right), -1\right\} + \delta, 0\right\}
$$

to ensure (4.21) and prevent $\mu^{(t)} > 0$, which may provide monotonic but slower convergence than $\mu^{(t)} = 0$. Since only the largest eigenvalue of $\boldsymbol{H}_M^{(t)}(\boldsymbol{L}^{(t)} + \boldsymbol{I})$ incorporates in the inequality (4.21), determining $\mu^{(t)}$ takes less computational time than solving the CARE.

4.4 Experiments

4.4.1 Experimental Settings

We evaluate performance of the learning methods for full-rank DPPs through experiments on synthetic and real-world datasets. For references, we take the fixed-point algorithm

(FP) [Mariet and Sra, 2015] and Adam [Kingma and Ba, 2015] as a representative gradient-based method. For Adam, we factorize the kernel matrix as $\boldsymbol{L} = \boldsymbol{V}\boldsymbol{V}^\top$ by $\boldsymbol{V} \in \mathbb{R}^{N \times N}$ and optimize $\boldsymbol{V}$ with the low-rank DPPs [Gartrell et al., 2017, Osogami et al., 2018]. We adopt full-batch learning for all the algorithms.

We provide the following two initialization schemes with reference to [Mariet and Sra, 2015]:

- `WISHART`: We sample an initial value from the Wishart distribution as $\boldsymbol{L}^{(0)} \sim \mathcal{W}(N, \boldsymbol{I})/N$.
- `BASIC`: We uniformly sample $v_{ij}^{(0)} \sim \mathcal{U}(0, \sqrt{2}/N)$ for $i, j = 1, 2, \ldots, N$ and initialize as $\boldsymbol{L}^{(0)} = \boldsymbol{V}^{(0)}\boldsymbol{V}^{(0)\top}$.

The `WISHART` initialization provides a near-identity matrix, while `BASIC` provides a unstructured matrix for $\boldsymbol{L}^{(0)}$.

We adopt the acceleration schemes for each algorithm. We set the step size $a = 1.3$ for the fixed-point algorithm[1] and the tolerance $\delta = 0.15$ for the proposed MM algorithm. For $T_{\mathrm{acc}} < t$, we use the default parameter $a = 1$ for the fixed-point algorithm, which monotonically increases the objective but no acceleration is applied, and the same way is used for the proposed MM. In Adam optimization, we employ the default values $\beta_1 = 0.999, \beta_2 = 0.9$ for the decay rates, and the machine epsilon $\epsilon = 10^{-8}$. The acceleration steps T_{acc} of the fixed-point and MM algorithms and the learning rate η of Adam are set to be different with the initialization schemes: $T_{\mathrm{acc}} = 5, \eta = 0.1$ for `WISHART` initialization and $T_{\mathrm{acc}} = 10, \eta = 0.01$ for `BASIC` initialization.

In each experiment, we stop learning when the criterion $\frac{|f(\boldsymbol{L}^{(t)}) - f(\boldsymbol{L}^{(t-1)})|}{|f(\boldsymbol{L}^{(t-1)})|} \leq \delta_{\mathrm{tol}}$ is satisfied. We set $\delta_{\mathrm{tol}} = 10^{-4}$ as the relative tolerance for all the experiments reported below. We implemented all the experiments in Julia, and all our experiments were run on a Linux Mint system with 32GB of RAM and an Intel Core i9-10900K CPU @ 3.70GHz.

4.4.2 Datasets

We compare the learning algorithms with the following three datasets.

[1] This is a possibly large value that does not fail optimization in our datasets.

Synthetic

We make true parameters as $\boldsymbol{L}^* = \boldsymbol{V}^*\boldsymbol{V}^{*\top}$ with $v^*_{ij} \sim \mathcal{U}(0, 10/N)$ for $i, j = 1, 2, \ldots, N$, and sample M realizations from the DPP $P_{\boldsymbol{L}^*}(\cdot)$. We consider three different problem sizes: $(N, M) = (32, 2{,}500)$, $(N, M) = (32, 10{,}000)$, and $(N, M) = (128, 2{,}500)$. Because the true parameters are constructed from the uniform distribution, they are likely to have no clear structure. Using this `Synthetic` dataset, we test the general applicability of our method.

In `Synthetic`, true parameters $\boldsymbol{L}^*$ are available; we assess goodness of estimation using not only log-likelihoods but also the von Neumann divergences $D_{\mathrm{vN}}(\boldsymbol{L}, \boldsymbol{L}^*) = \mathrm{tr}(\boldsymbol{L}\log\boldsymbol{L} - \boldsymbol{L}\log\boldsymbol{L}^* - \boldsymbol{L} + \boldsymbol{L}^*)$, which is a Bregman divergence for positive definite matrices.

Nottingham

We apply our method to the `Nottingham` music dataset[2], which was used in [Boulanger-Lewandowski et al., 2012, Osogami et al., 2018]. The dataset contains more than 1,000 folk tracks in the ABC format in which a sequence of chords is stored. We treat each chord in the tracks as an i.i.d. sample of a DPP on the ground set $\{1, 2, \ldots, 88\}$, where $N = 88$ is the number of keys. We randomly pick 25 tracks and that yields $M = 6{,}364$ samples on average.

In `Nottingham`, there is large disparity in the probability of each item appearing, with very low- and high-pitched keys being rarely used. Moreover, music theory prohibits certain key combinations within a chord. From these facts, the optimal $\boldsymbol{L}^*$ of the `Nottingham` dataset is expected to have unknown but particular structure.

Amazon Baby Registry

`Amazon baby registry` has served as a benchmark for learning methods of DPPs since [Gillenwater et al., 2014]. It contains 13 categories of child care products, including “feeding” and “carseats,” and on average, has $N = 71$ items and $M = 8{,}585$ samples, respectively. We run our experiment on each of the 13 categories to assess performance of the learning methods for medium-sized recommender systems.

[2] Available at https://abc.sourceforge.net/NMD/.

Table 4.1: Final mean log-likelihoods, runtimes, and von Neumann divergences $D_{\mathrm{vN}}(\boldsymbol{L}, \boldsymbol{L}^*)$ of the `Synthetic` datasets. Each value is computed from the average or standard deviation of 30 trials with the accelerated settings.

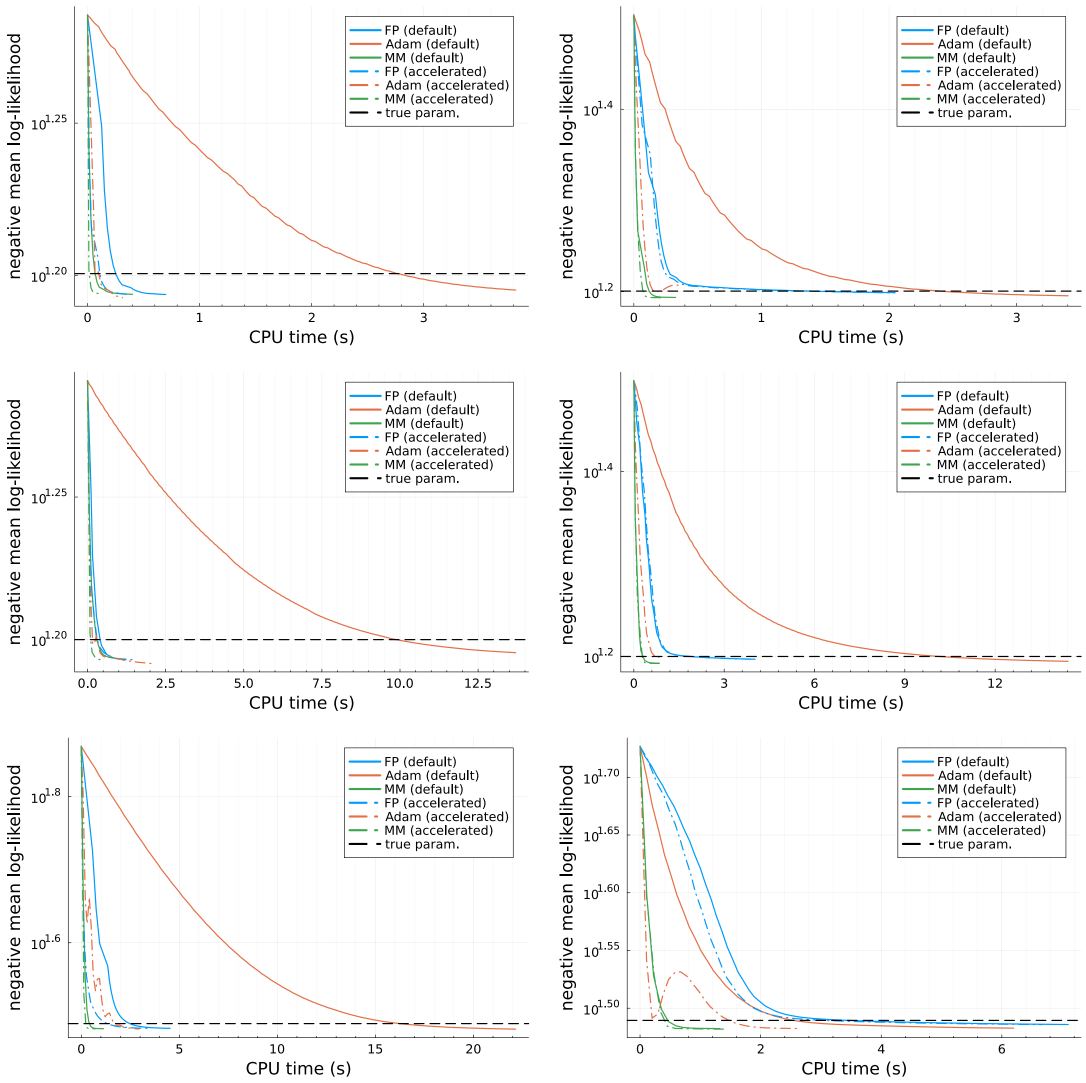
		WISHART			BASIC		
Data Size	Method	Log-likelihood	Runtime (s)	vN div.	Log-likelihood	Runtime (s)	vN div.
$N = 32$ $M = 2{,}500$	FP	−15.58 ± 0.15	0.39 ± 0.03	**38.22** ± 1.85	−15.61 ± 0.20	1.46 ± 0.31	41.39 ± 4.17
	Adam	**−15.55** ± 0.16	0.36 ± 0.23	56.77 ± 9.40	−15.64 ± 0.34	0.71 ± 0.37	33.42 ± 3.13
	MM	−15.58 ± 0.15	**0.18** ± 0.07	42.63 ± 2.38	**−15.45** ± 0.20	**0.21** ± 0.03	**30.16** ± 1.97
$N = 32$ $M = 10{,}000$	FP	**−15.58** ± 0.17	1.32 ± 0.14	**38.05** ± 2.01	−15.71 ± 0.14	5.59 ± 0.85	40.79 ± 3.29
	Adam	**−15.58** ± 0.17	1.23 ± 0.56	49.35 ± 4.54	−15.70 ± 0.22	3.05 ± 1.45	32.50 ± 2.11
	MM	**−15.58** ± 0.18	**0.48** ± 0.09	42.53 ± 2.43	**−15.55** ± 0.14	**0.77** ± 0.09	**29.75** ± 1.52
$N = 128$ $M = 2{,}500$	FP	−30.14 ± 0.18	3.36 ± 0.22	**36.17** ± 0.45	−30.34 ± 0.19	6.37 ± 0.48	52.62 ± 1.74
	Adam	−30.18 ± 0.22	2.50 ± 0.40	44.88 ± 1.53	−30.46 ± 1.08	2.30 ± 0.56	39.44 ± 5.54
	MM	**−30.11** ± 0.18	**0.69** ± 0.05	42.54 ± 0.53	**−30.08** ± 0.19	**1.27** ± 0.21	**32.15** ± 0.55

4.4.3 Experimental Results

Synthetic

The final mean log-likelihoods, runtimes, and von-Neumann divergence values of the `Synthetic` datasets with the acceleration are presented in Table 4.1. For each experiment, we conducted 30 trials with different $\boldsymbol{L}^*$ and $\boldsymbol{L}^{(0)}$ and calculated the average and standard deviation. As shown in Table 4.1, our method (MM) achieves the best runtimes for all the settings. While the final log-likelihood values are almost equivalent by the algorithms in `WISHART` initialization, those obtained by the proposed MM tend to be larger in `BASIC` initialization. Furthermore, our method also produces the best von Neumann divergences D_{vN} with `BASIC` initialization and moderately performs with `WISHART` initialization. The results show good stablity of our method; the proposed algorithm is considered to be favorable in standard situations. The result of the `Synthetic` datasets without the acceleration is also shown in Section 4.8.

In Figure 4.3, we show the learning curves with and without acceleration. While the fixed-point algorithm convergences stably yet slightly slow without the acceleration, the accelerated version becomes competitive in `WISHART` initialization. The Adam optimizer may temporarily fall into poor local optima, depending on the initial value. On the other hand, the proposed MM algorithm consistently indicates stable and rapid convergence both with and without the acceleration.



(a) WISHART

(b) BASIC

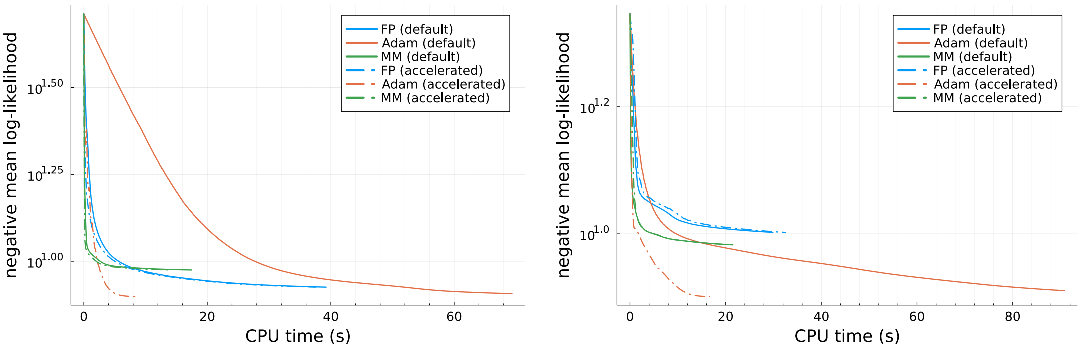
Figure 4.3: Learning curves of the Synthetic datasets. Top: $(N, M) = (32, 2{,}500)$, Medium: $(N, M) = (32, 10{,}000)$, Bottom: $(N, M) = (128, 2{,}500)$. Results with the default parameters ($T_{\text{acc}} = 0$ for fixed-point and MM, and $\eta = 0.001$ for Adam) are also shown.

Nottingham

The results of the Nottingham dataset with the acceleration are presented in Table 4.2, and the learning curves with and without acceleration are showed in Figure 4.4. The convergence of Adam is remarkably rapid in the Nottingham dataset.

Table 4.2: Final mean log-likelihoods and runtimes of the `Nottigham` dataset. Each value is computed from the average or standard deviation of 30 trials with the accelerated settings.

	`WISHART`		`BASIC`	
Method	Log-likelihood	Runtime (s)	Log-likelihood	Runtime (s)
FP	-8.31 ± 0.22	40.68 ± 2.86	-10.13 ± 0.27	33.19 ± 8.08
Adam	$\mathbf{-7.84} \pm 1.02$	$\mathbf{9.01} \pm 1.59$	$\mathbf{-7.92} \pm 0.64$	$\mathbf{21.05} \pm 6.75$
MM	-9.51 ± 0.24	19.69 ± 4.89	-9.58 ± 0.21	19.11 ± 5.49



(a) `WISHART`

(b) `BASIC`

Figure 4.4: Learning curves of the `Nottingham` dataset. Results with the default parameters ($T_{\text{acc}} = 0$ for fixed-point and MM, and $\eta = 0.001$ for Adam) are also shown.

Under the `BASIC` initialization, the fixed-point and MM algorithms get stuck in poor local optima. Since the optimal $\boldsymbol{L}^*$ is considered to have a particular structure, the `BASIC` initialization may not be compatible with `Nottingham`. We can also find the acceleration scheme of the MM algorithm does not perform well in Figure 4.4 (see also the result without the acceleration shown in Section 4.8). This may be because the assumption $\bigcup_{m=1}^{M} \mathcal{A}_m = \mathcal{Y}$ for the accelerated MM is not satisfied in the `Nottingham` dataset.

Amazon Baby Registry

In Table 4.3, we show the results with the accelerated algorithms in all the 13 categories of `Amazon baby registry`. Overall, our algorithm achieves moderately better log-likelihood values and outstanding convergence speeds in most categories. Adam

Table 4.3: Final mean log-likelihoods and runtimes of the `Amazon baby registry` dataset. Each value is computed from the average or standard deviation of 30 trials with the accelerated settings and initialized by `WISHART`.

Category	Method	Log-likelihood	Runtime (s)	Category	Method	Log-likelihood	Runtime (s)
Apparel	FP	−10.20 ± 0.00	24.54 ± 0.69	Gear	FP	−9.27 ± 0.00	30.54 ± 0.90
$N = 100$	Adam	**−10.08** ± 0.26	17.99 ± 2.61	$N = 100$	Adam	**−9.16** ± 0.41	25.85 ± 5.74
$M = 14{,}970$	MM	−10.17 ± 0.00	**3.13** ± 0.30	$M = 16{,}823$	MM	−9.24 ± 0.00	**2.02** ± 0.38
Bath	FP	−8.79 ± 0.00	26.51 ± 0.47	Health	FP	−7.59 ± 0.00	13.22 ± 0.35
$N = 100$	Adam	**−8.72** ± 0.79	17.97 ± 4.84	$N = 62$	Adam	**−7.37** ± 0.27	10.06 ± 1.66
$M = 14{,}542$	MM	−8.75 ± 0.00	**2.06** ± 0.47	$M = 14{,}057$	MM	−7.55 ± 0.00	**2.16** ± 0.44
Bedding	FP	−8.79 ± 0.00	32.23 ± 0.73	Media	FP	−8.56 ± 0.00	4.01 ± 0.67
$N = 100$	Adam	**−8.59** ± 0.18	23.26 ± 1.31	$N = 58$	Adam	**−8.39** ± 0.16	2.97 ± 1.07
$M = 16{,}370$	MM	−8.77 ± 0.00	**4.79** ± 1.10	$M = 5{,}904$	MM	−8.52 ± 0.01	**1.75** ± 0.75
Carseats	FP	−5.18 ± 0.06	5.04 ± 3.27	Safety	FP	−4.76 ± 0.16	8.93 ± 7.46
$N = 34$	Adam	**−4.82** ± 0.29	**2.03** ± 0.33	$N = 36$	Adam	**−4.30** ± 0.00	**2.28** ± 0.10
$M = 7{,}566$	MM	−5.00 ± 0.05	4.96 ± 1.45	$M = 8{,}892$	MM	−4.57 ± 0.05	6.19 ± 2.10
Diaper	FP	−10.71 ± 0.00	27.16 ± 0.83	Strollers	FP	−5.66 ± 0.06	4.58 ± 3.21
$N = 100$	Adam	**−10.61** ± 0.35	25.75 ± 5.96	$N = 40$	Adam	**−5.25** ± 0.38	**2.35** ± 0.39
$M = 16{,}759$	MM	−10.67 ± 0.00	**3.21** ± 0.53	$M = 7{,}393$	MM	−5.46 ± 0.05	6.12 ± 2.39
Feeding	FP	−12.17 ± 0.00	28.97 ± 0.36	Toys	FP	−8.10 ± 0.00	7.65 ± 0.71
$N = 100$	Adam	−12.17 ± 0.27	18.38 ± 5.11	$N = 62$	Adam	**−7.94** ± 0.27	5.77 ± 1.25
$M = 19{,}001$	MM	**−12.15** ± 0.00	**3.11** ± 0.39	$M = 10{,}073$	MM	−8.07 ± 0.00	**1.45** ± 0.34
Furniture	FP	−4.86 ± 0.13	4.93 ± 4.75				
$N = 32$	Adam	**−4.40** ± 0.00	**1.88** ± 0.05				
$M = 7{,}093$	MM	−4.65 ± 0.05	5.37 ± 1.78				

tends to produce the best final log-likelihoods but they are not statistically significant in most cases. Especially, when the sample size is relatively large, such as $M > 10{,}000$, our algorithm outperforms in the convergence speed that is about 5-10 times faster than the fixed-point algorithm.

Although the convergences of the MM algorithm seems to be slow in some of the smaller categories in Table 4.3, that is not very serious. In these cases, the MM algorithm quickly reaches a near optimum value, but takes longer to meet the stopping criterion. By managing the stopping criterion, we may be able to stop its learning much earlier.

4.5 Discussion

In this chapter, we developed an efficient learning method for full-rank DPPs based on the MM algorithm. Compared with the existing methods, our algorithm has many advantages: it has guaranteed convergence and monotonicity, requires no bothersome hyperparameters, convergences rapidly and stably, and is easy to implement. Upon considering the performance of our algorithm, we revealed that our algorithm provides a locally tighter minorizer than the existing method. We also assessed the empirical performance of our method through experiments on both synthetic and real-world datasets, outperforming in terms of convergence speed and reaching a better estimate in most experimental settings.

4.6 Proof of Proposition 4.3

4.6.1 Derivation of Equation (4.15)

In (4.14), the constant term is given by

$$\begin{aligned}\xi(\boldsymbol{L}^{(t)}) = \frac{1}{M}\sum_{m=1}^{M}&\left\{\log\det(\boldsymbol{U}_{\mathcal{A}_m}\boldsymbol{L}^{(t)}\boldsymbol{U}_{\mathcal{A}_m}^\top) + |\mathcal{A}_m|\right\}\\&+\log\det\{(\boldsymbol{L}^{(t)}+\boldsymbol{I})^{-1}\boldsymbol{L}^{(t)}\} + \mathrm{tr}\{(\boldsymbol{L}^{(t)}+\boldsymbol{I})^{-1}\}.\end{aligned}$$

By the following inequality from the Taylor expansion

$$-\log\det(\boldsymbol{L}^{(t)}) \geq -\log\det(\boldsymbol{L}) - \mathrm{tr}\{\boldsymbol{L}^{-1}(\boldsymbol{L}^{(t)}-\boldsymbol{L})\},$$

we have

$$\begin{aligned}g(\boldsymbol{L}|\boldsymbol{L}^{(t)}) - h(\boldsymbol{L}|\boldsymbol{L}^{(t)}) &= \mathrm{tr}\{(\boldsymbol{L}^{(t)}+\boldsymbol{I})^{-1}(\boldsymbol{L}^{-1}\boldsymbol{L}^{(t)} - \boldsymbol{I} - \boldsymbol{L} + \boldsymbol{L}^{(t)})\}\\&\quad+\log\det(\boldsymbol{L}) - \log\det(\boldsymbol{L}^{(t)})\\&\geq \mathrm{tr}\{(\boldsymbol{L}^{(t)}+\boldsymbol{I})^{-1}(\boldsymbol{L}^{-1}\boldsymbol{L}^{(t)} - \boldsymbol{I} - \boldsymbol{L} + \boldsymbol{L}^{(t)})\} - \mathrm{tr}\{\boldsymbol{L}^{-1}(\boldsymbol{L}^{(t)}-\boldsymbol{L})\}\\&= \mathrm{tr}\{(\boldsymbol{L}^{(t)}+\boldsymbol{I})^{-1}(\boldsymbol{L}^{-1}\boldsymbol{L}^{(t)} - \boldsymbol{L} + 2\boldsymbol{L}^{(t)})\} - \mathrm{tr}\{\boldsymbol{L}^{-1}\boldsymbol{L}^{(t)}\} - N + N\\&= \mathrm{tr}\{(\boldsymbol{L}^{(t)}+\boldsymbol{I})^{-1}(2\boldsymbol{L}^{(t)} - \boldsymbol{L} - \boldsymbol{L}^{(t)}\boldsymbol{L}^{-1}\boldsymbol{L}^{(t)})\}.\end{aligned}$$

4.6.2 Derivation of Equation (4.16)

Let $\boldsymbol{L} = \boldsymbol{L}^{(t)} + \delta \boldsymbol{M}$, where $\delta > 0$ is a sufficiently small coefficient and $\boldsymbol{M}$ is a symmetric matrix whose all eigenvalues are in $[-1, 1]$. Then, we can approximate the matrix inverse as $\boldsymbol{L}^{-1} = (\boldsymbol{L}^{(t)} + \delta\boldsymbol{M})^{-1} \approx \boldsymbol{L}^{(t)-1} - \delta\boldsymbol{L}^{(t)-1}\boldsymbol{M}\boldsymbol{L}^{(t)-1}$ by the Taylor expansion. Using this approximation, we have

$$\begin{aligned} 2\boldsymbol{L}^{(t)} - \boldsymbol{L} - \boldsymbol{L}^{(t)}\boldsymbol{L}^{-1}\boldsymbol{L}^{(t)} &= 2\boldsymbol{L}^{(t)} - (\boldsymbol{L}^{(t)} + \delta\boldsymbol{M}) - \boldsymbol{L}^{(t)}(\boldsymbol{L}^{(t)} + \delta\boldsymbol{M})^{-1}\boldsymbol{L}^{(t)} \\ &\approx 2\boldsymbol{L}^{(t)} - (\boldsymbol{L}^{(t)} + \delta\boldsymbol{M}) - \boldsymbol{L}^{(t)}(\boldsymbol{L}^{(t)-1} - \delta\boldsymbol{L}^{(t)-1}\boldsymbol{M}\boldsymbol{L}^{(t)-1})\boldsymbol{L}^{(t)} \\ &= \boldsymbol{O}. \end{aligned}$$

4.7 Proof of Proposition 4.4

Proof. $f_{\mu^{(t)}}(\boldsymbol{L}|\boldsymbol{L}^{(t)})$ can be minorized as:

$$\begin{aligned} &f_{\mu^{(t)}}(\boldsymbol{L}|\boldsymbol{L}^{(t)}) \\ &= \frac{1}{M}\sum_{m=1}^{M}(\log\det([\boldsymbol{L}]_{\mathcal{A}_m}) - \underbrace{\mu^{(t)}\log\det([\boldsymbol{L}]_{\mathcal{A}_m})}_{\substack{\text{majorizing by (4.3) w/} \\ \boldsymbol{X} \to [\boldsymbol{L}]_{\mathcal{A}_m}, \\ \boldsymbol{Y} \to [\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}}} - \mu^{(t)}\mathrm{tr}([\boldsymbol{L}]_{\mathcal{A}_m}^{-1}[\boldsymbol{L}^{(t)}]_{\mathcal{A}_m})) \\ &\quad - \underbrace{\log\det(\boldsymbol{L} + \boldsymbol{I})}_{\substack{\text{majorizing by (4.3) w/} \\ \boldsymbol{X} \to \boldsymbol{L} + \boldsymbol{I}, \\ \boldsymbol{Y} \to \boldsymbol{L}^{(t)} + \boldsymbol{I}}} + \text{const.} \\ &\geq \frac{1}{M}\sum_{m=1}^{M}(\underbrace{\log\det([\boldsymbol{L}]_{\mathcal{A}_m})}_{\substack{\text{minorizing by (4.4) w/} \\ \boldsymbol{X} \to [\boldsymbol{L}]_{\mathcal{A}_m}, \\ \boldsymbol{Y} \to [\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}}} - \mu^{(t)}\mathrm{tr}([\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}^{-1}[\boldsymbol{L}]_{\mathcal{A}_m}) - \mu^{(t)}\mathrm{tr}([\boldsymbol{L}]_{\mathcal{A}_m}^{-1}[\boldsymbol{L}^{(t)}]_{\mathcal{A}_m})) \\ &\quad - \mathrm{tr}\{(\boldsymbol{L} + \boldsymbol{I})^{-1}\boldsymbol{L}\} + \text{const.} \end{aligned}$$

$$
\begin{aligned}
&\geq -\frac{1}{M}\sum_{m=1}^{M}((1+\mu^{(t)}) \underbrace{\mathrm{tr}([\boldsymbol{L}]_{\mathcal{A}_m}^{-1}[\boldsymbol{L}^{(t)}]_{\mathcal{A}_m})}_{\substack{\text{majorizing by (4.7) w/}\\ \boldsymbol{APA}^\top \to [\boldsymbol{L}]_{\mathcal{A}_m},\\ \boldsymbol{S} \to [\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}}} + \mu^{(t)}\mathrm{tr}([\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}^{-1}[\boldsymbol{L}]_{\mathcal{A}_m})) \\
&\quad - \mathrm{tr}\{(\boldsymbol{L}+\boldsymbol{I})^{-1}\boldsymbol{L}\} + \mathrm{const.} \\
&\geq -\frac{1+\mu^{(t)}}{M}\sum_{m=1}^{M}\mathrm{tr}(\boldsymbol{L}^{(t)}\boldsymbol{U}_{\mathcal{A}_m}^\top[\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}^{-1}\boldsymbol{U}_{\mathcal{A}_m}\boldsymbol{L}^{(t)}\boldsymbol{L}^{-1}) \\
&\quad - \frac{\mu^{(t)}}{M}\sum_{m=1}^{M}\mathrm{tr}([\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}^{-1}[\boldsymbol{L}]_{\mathcal{A}_m}) - \mathrm{tr}\{(\boldsymbol{L}^{(t)}+\boldsymbol{I})^{-1}\boldsymbol{L}\} + \mathrm{const.} \\
&= g_{\mu^{(t)}}(\boldsymbol{L}|\boldsymbol{L}^{(t)}).
\end{aligned}
$$

□

4.8 Additional Experimental Results

Table 4.4 shows the learning result of the `Synthetic` dataset with the default (non-accelerated) settings. We find that the proposed MM algorithm with the default setting still performs better than the other algorithms with the accelerated settings, shown in Table 4.1.

Table 4.5 shows the result of the `Nottingham` dataset with the default settings. In contrast to `Synthetic`, the performance of the MM algorithm with and without the acceleration is not much different (cf. Table 4.2). This may be due to the absence of the assumption required in the accelerated MM algorithm. We need $\bigcup_m \mathcal{A}_m = \mathcal{Y}$ in Section 4.3, but `Nottingham` does not satisfy that as described in Section 4.4.

4.9 Mode Structure of Log-likelihood

While we aimed to reach a local optimum of the (mean) log-likelihood (4.1), the global structure of the objective may be interested and informative. To seek the modal structure of the objective, we define a Bayesian model for full-rank DPPs with weakly informative priors and explore the posterior with a Markov chain Monte Carlo (MCMC) method. We

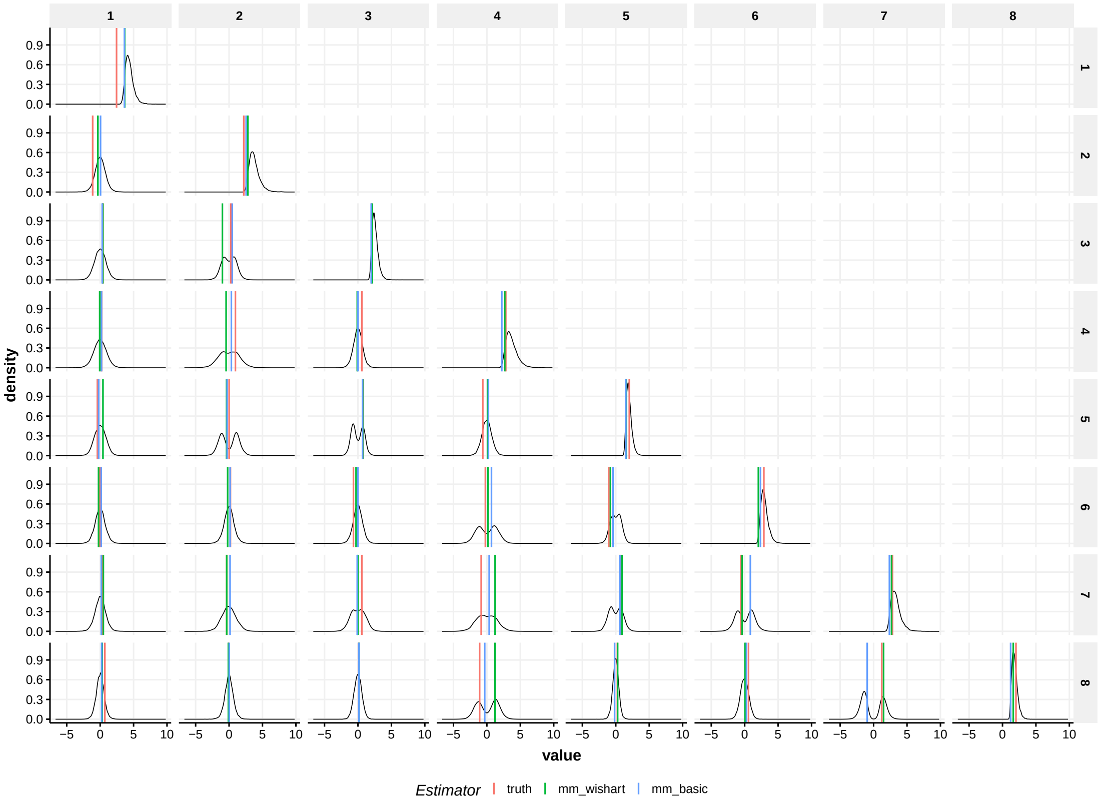


Figure 4.5: Posterior marginals of L_{ij} for $i, j = 1, \ldots, 8$ approximated by MCMC samples. The vertical lines show the ground truth and point estimates by the proposed MM with `WISHART` and `BASIC` initialization schemes.

Table 4.4: Final mean log-likelihoods, runtimes, and von Neumann divergences $D_{\mathrm{vN}}(\boldsymbol{L}, \boldsymbol{L}^*)$ of the `Synthetic` datasets. Each value is computed from the average or standard deviation of 30 trials with the non-accelerated settings.

		`WISHART`			`BASIC`		
Data Size	Method	Log-likelihood	Runtime (s)	vN div.	Log-likelihood	Runtime (s)	vN div.
$N = 32$ $M = 2{,}500$	FP	−15.58 ± 0.15	0.43 ± 0.06	38.20 ± 1.84	−15.61 ± 0.21	1.69 ± 0.22	42.19 ± 4.19
	Adam	−15.63 ± 0.15	3.29 ± 0.25	63.07 ± 5.71	−15.54 ± 0.20	3.41 ± 0.17	29.51 ± 1.85
	MM	−15.58 ± 0.15	0.40 ± 0.06	43.94 ± 2.62	−15.46 ± 0.20	0.32 ± 0.03	30.15 ± 1.99
$N = 32$ $M = 10{,}000$	FP	−15.58 ± 0.18	1.32 ± 0.17	38.03 ± 2.00	−15.72 ± 0.14	5.53 ± 0.96	41.61 ± 3.30
	Adam	−15.66 ± 0.17	12.83 ± 1.11	61.41 ± 5.40	−15.63 ± 0.14	14.59 ± 0.56	29.08 ± 1.23
	MM	−15.58 ± 0.18	1.22 ± 0.16	43.95 ± 2.58	−15.56 ± 0.14	1.00 ± 0.11	29.74 ± 1.47
$N = 128$ $M = 2{,}500$	FP	−30.14 ± 0.18	3.70 ± 0.22	36.20 ± 0.44	−30.35 ± 0.19	6.56 ± 0.45	53.47 ± 1.79
	Adam	−30.05 ± 0.19	24.05 ± 0.78	85.47 ± 4.02	−30.12 ± 0.19	5.84 ± 0.23	33.44 ± 0.56
	MM	−30.11 ± 0.18	1.39 ± 0.08	44.68 ± 0.62	−30.10 ± 0.19	1.29 ± 0.07	32.26 ± 0.50

Table 4.5: Final mean log-likelihoods and runtimes of the `Nottigham` dataset. Each value is computed from the average or standard deviation of 30 trials with the non-accelerated settings.

	`WISHART`		`BASIC`	
Method	Log-likelihood	Runtime (s)	Log-likelihood	Runtime (s)
FP	−8.30 ± 0.22	40.92 ± 3.01	−10.14 ± 0.28	33.75 ± 6.84
Adam	−7.81 ± 0.25	68.12 ± 4.94	−8.02 ± 0.26	103.27 ± 15.57
MM	−9.51 ± 0.25	21.73 ± 6.04	−9.59 ± 0.22	19.95 ± 3.59

use the following Bayesian model:

$$\begin{aligned}
P(\mathcal{A}_m|\boldsymbol{L}) &= P_{\boldsymbol{L}}(\mathcal{A}_m) && \text{for} \quad m = 1, \ldots, M, \\
\boldsymbol{L} &= \operatorname{diag}(\sigma_1, \ldots, \sigma_N)\ \Omega\ \operatorname{diag}(\sigma_1, \ldots, \sigma_N), \\
p(\Omega) &= \mathrm{LKJ}(\eta), \\
p(\sigma_n) &= \mathrm{Cauchy}_+(\gamma), && \text{for} \quad n = 1, \ldots, N,
\end{aligned}$$

where $\mathrm{LKJ}(\eta) \propto (\det \Omega)^{\eta-1}$ denotes the Lewandowski–Kurowicka–Joe (LKJ) distribution [Lewandowski et al., 2009] and $\mathrm{Cauchy}_+(\gamma)$ denotes the half-Cauchy distribution with the half-width at half-maximum (HWHM) parameter $\gamma > 0$. We use the hyperparameters $\eta = 1.001$ and $\gamma = 100$ which ensure weakly informative priors, and obtain 15,000 MCMC samples[3] by the No-U-Turn sampler (NUTS) [Hoffman and Gelman, 2014,

[3] Without the burn-in periods.

Betancourt, 2018] implemented on Stan [Carpenter et al., 2017]. We generate a synthetic dataset consisting of $M = 1{,}000$ samples from a DPP with the ground truth parameter $\boldsymbol{L}^* = \boldsymbol{V}^* \boldsymbol{V}^{*\top}$, where $v_{ij} \sim \mathcal{U}(-10, 10)/16$ for $i = 1, \ldots, 8,\ j = 1, \ldots, 16$ (i.e., the size of the ground set is $N = |\mathcal{Y}| = 8$).

Figure 4.5 shows the marginal posterior densities $p(L_{ij} | \mathcal{A}_1, \ldots, \mathcal{A}_M)$ for $i, j = 1, \ldots, 8$ and the point estimates obtained by the proposed MM algorithm. Notably the non-diagonal elements of L_{ij} are often bimodal with symmetry around the origin and the MM reasonably reaches the local optima. As pointed out in [Kulesza, 2012, Section 4.3.1], the kernel matrix $\boldsymbol{L}$ in the likelihood of DPPs (2.15) is unidentifiable because the map $\mathcal{M}_{\boldsymbol{D}} : \boldsymbol{L} \mapsto \boldsymbol{DLD}$ with a diagonal matrix $\boldsymbol{D}$ such that $D_{ii} \in \{-1, +1\}$ for $i = 1, \ldots, N$ does not change the likelihood value: $\forall \mathcal{A} \subseteq \mathcal{Y},\ P_{\boldsymbol{L}}(\mathcal{A}) = P_{\mathcal{M}_{\boldsymbol{D}}(\boldsymbol{L})}(\mathcal{A})$. The symmetric bimodality shown in Figure 4.5 is led by this unidentifiability.

5 Conclusion

In this dissertation, we focused on probabilistic models characterized by a kernel matrix and their learning algorithms. Specifically, we developed a GP-based generative model of KMD and an efficient algorithm for learning the full-rank kernel matrix of DPPs.

In Chapter 3, we proposed a Bayesian generative model of KMD based on an unsupervised GP, and named it GPKMD (which stands for Gaussian process Koooman mode decomposition). The derivation of the GPKMD likelihood was somewhat similar to the GP regression, as introduced in Section 2.2. This involved the marginalization of a countably infinite-dimensional coefficient vector within the mean vector. That results complex normal distributions of which a very high-dimensional vector, and whose covariance matrices follow a "diagonal + Kronecker factorizable" structure. GPs with such covariance matrices are called Kronecker GPs, and we developed a faster evaluation method of the likelihood and its derivatives than the existing method [Stegle et al., 2011]. Notably, we can estimate the latent variables of KMD $\{\boldsymbol{x}_t\}$ due to the generative modeling. Our GPKMD is the first to address direct estimation of the latent variables within the context of KMD. In Section 3.5, we applied GPKMD to both synthetic and real datasets and interpreted the results from various aspects through the estimated Koopman eigenvalues $\{\lambda_k\}$, Koopman modes $\{\boldsymbol{w}_k\}$, and latent variables $\{\boldsymbol{x}_t\}$.

This study has some limitations and future work is suggested. In this work, we did not show the estimated eigenfunctions $\{\phi_k\}$. The eigenfunctions are implicitly determined by the kernel function and the latent variables in our model, but their explicit estimates are intractable. There is also difficulty in learning the Koopman eigenvalues $\{\lambda_k\}$. The angle of the k-th eigenvalue, $\arg \lambda_k$, corresponds to the frequency of the k-th mode. In (3.12), however, the eigenvalues are included in the form $\Lambda\Lambda^* = \mathrm{diag}(|\lambda_k|^2)$; hence, the angles $\{\arg \lambda_k\}$ do not affect the likelihood. In addition, the gradient of the likelihood (3.12) w.r.t. λ_k is proportional to λ_k itself, and the angle remains fixed during gradient-based learning. In the examples in Section 3.5, we practically use the DMD estimates of $\{\lambda_k\}$ to alleviate this difficulty. One promising approach for estimating $\{\phi_k\}$ and $\{\lambda_k\}$ is to use approximated GPs with finite-dimensional features, such as random Fourier features (RFFs) [Rahimi and Recht, 2007]. Learning GPs with RFFs reduces to that of a Bayesian linear function model, making it possible to obtain explicit expressions of the estimated eigenfunctions and the eigenvalues angles. Furthermore, while we employed gradient-based MAP estimation in Section 3.5, credible interval estimation of GPKMD could provide more informative results. The sparse variational Gaussian process (SVGP) is a well-established variational Bayesian method for learning GPs, which maximizes the evidence lower-bound (ELBO) instead of the marginalized posterior [Titsias, 2009, Titsias and Lawrence, 2010]. Although Wild et al. [2021] has explored connections between the Nyström method and SVGP, the development of a variational inference method for GPKMD remains a potential area for future work.

In Chapter 4, we proposed a learning algorithm that solves MLE for full-rank DPPs on a finite ground set. Using the MM algorithm, MLE for DPPs was interestingly transformed into a iterative process of solving CAREs, which belong to a special class of quadratic matrix equations. In addition, we developed a generalization of our algorithm for further acceleration. We also conducted a theoretical comparison between our algorithm and the existing method [Mariet and Sra, 2015] in the sense of the tightness of the minorizers. One notable feature of our accelerated algorithm is its ability to adaptively determine the step size in each iteration, whereas the existing method [Mariet and Sra, 2015] uses fixed step size hyperparameters. As demonstrated in the experiments in Section 4.4, our algorithm performs the best runtimes across a variety of settings.

We believe that our algorithm is a strong candidate for learning full-rank DPPs at the present moment, but there is still much future work to be considered. First, we need to deepen our understanding of the performance of our method. Proposition 4.3 partially addresses this question, but it is just an implication. One considerable future direction is to establish general recipes for comparing MM with different minorizers. Second, scaling up our method for large N is a crucial issue. Several numerical algorithms for solving large-sized CARE (4.12) have been proposed based on some structure of a problem: low-rank structure and/or sparsity [Bini et al., 2011, Simoncini, 2016]. On the other hand, our CARE (4.9) formed by full-rank and dense matrices, therefore, exploring a good CARE solver is considered to be an essential task.

More broadly, there are many open problems about learning DPPs. For example, incorporating sparsity into the kernel matrix $\boldsymbol{L}$ potentially enhances interpretability and computational efficiency. Although studies have addressed the sparsity in the context of (inverse) covariance selection, such as graphical lasso [Banerjee et al., 2008, Friedman et al., 2008], any existing works have not addressed the sparsity of DPPs. The learning problem of DPPs on an infinite ground set is also open. While we focused on DPPs on a finite set in Chapter 4, DPPs on an infinite set become relevant when each item possesses a feature vector. For example, consider a fashion online store in which each item has a D-dimensional feature vector extracted from the image. In such cases, the purchasing behavior of customers could be modeled by a DPP on $\mathbb{R}^D$. Dupuy and Bach [2018] addressed the problem with Fourier bases, but the applicability to general problems is questionable because low-rank structure is strongly assumed. Multiple kernel learning [Gönen and Alpaydin, 2011] could offer a promising approach to this challenge. At a high level, generalizing DPPs to control negative (or positive) dependence is helpful for further development of random subset models. Although α-DPPs have been developed to bridge the behavior of bosons (having positive dependence) and fermions (having negative dependence) [Vere-Jones, 1997, Hough et al., 2006], they have a computational issue for applying to machine learning problems. We thus believe that a machine learning-compatible generalization of DPPs is a crucial step forward.

Bibliography

N. Anari, S. O. Gharan, and A. Rezaei. Monte Carlo Markov Chain Algorithms for Sampling Strongly Rayleigh Distributions and Determinantal Point Processes. In *Conference on Learning Theory*, pages 103–115. PMLR, June 2016.

N. Aronszajn. Theory of Reproducing Kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950. ISSN 0002-9947. doi: 10.2307/1990404.

O. Banerjee, L. E. Ghaoui, and A. d'Aspremont. Model Selection Through Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data. *Journal of Machine Learning Research*, 9(15):485–516, 2008. ISSN 1533-7928.

A. Beck and M. Teboulle. Mirror Descent and Nonlinear Projected Subgradient Methods for Convex Optimization. *Operations Research Letters*, 31(3):167–175, May 2003. ISSN 0167-6377. doi: 10.1016/S0167-6377(02)00231-6.

P. Benner and R. Byers. An Exact Line Search Method for Solving Generalized Continuous-Time Algebraic Riccati Equations. *IEEE Transactions on Automatic Control*, 43(1):101–107, Jan. 1998. ISSN 1558-2523. doi: 10.1109/9.654908.

A. Berlinet and C. Thomas-Agnan. Reproducing kernel Hilbert spaces in probability and statistics, 2004.

M. Betancourt. A Conceptual Introduction to Hamiltonian Monte Carlo, July 2018.

D. Bini, B. Iannazzo, and B. Meini. *Numerical Solution of Algebraic Riccati Equations*. Society for Industrial and Applied Mathematics, Philadelphia, 2011. ISBN 978-1-61197-208-5.

E. V. Bonilla, K. Chai, and C. Williams. Multi-Task Gaussian Process Prediction. *Advances in Neural Information Processing Systems*, 20, 2007.

J. Borcea, P. Brändén, and T. M. Liggett. Negative Dependence and the Geometry of Polynomials. *Journal of the American Mathematical Society*, 22(2):521–567, 2009. ISSN 0894-0347.

A. Borodin and E. M. Rains. Eynard–Mehta Theorem, Schur Process, and Their Pfaffian Analogs. *Journal of Statistical Physics*, 121(3):291–317, Nov. 2005. ISSN 1572-9613. doi: 10.1007/s10955-005-7583-z.

N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription. In *Proceedings of the 29th International Conference on Machine Learning*, ICML'12, pages 1881–1888, Madison, WI, USA, 2012. Omnipress. ISBN 978-1-4503-1285-1.

D. R. Burt, C. E. Rasmussen, and M. van der Wilk. Convergence of Sparse Variational Inference in Gaussian Processes Regression. *Journal of Machine Learning Research*, 21(131):1–63, 2020. ISSN 1533-7928.

B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: A Probabilistic Programming Language. *Journal of Statistical Software*, 76:1–32, Jan. 2017. ISSN 1548-7660. doi: 10.18637/jss.v076.i01.

L. Chen, G. Zhang, and E. Zhou. Fast Greedy MAP Inference for Determinantal Point Process to Improve Recommendation Diversity. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

M. J. Colbrook and A. Townsend. Rigorous data-driven computation of spectral properties of Koopman operators for dynamical systems. *Communications on Pure and Applied Mathematics*, 77(1):221–283, 2024. ISSN 1097-0312. doi: 10.1002/cpa.22125.

M. J. Colbrook, L. J. Ayton, and M. Szőke. Residual dynamic mode decomposition: Robust and verified Koopmanism. *Journal of Fluid Mechanics*, 955:A21, Jan. 2023. ISSN 0022-1120, 1469-7645. doi: 10.1017/jfm.2022.1052.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sept. 1995. ISSN 1573-0565. doi: 10.1007/BF00994018.

A. C. Damianou, M. K. Titsias, and N. D. Lawrence. Variational Inference for Latent Variables and Uncertain Inputs in Gaussian Processes. *Journal of Machine Learning Research*, 17(42):1–62, 2016. ISSN 1533-7928.

S. T. M. Dawson, M. S. Hemati, M. O. Williams, and C. W. Rowley. Characterizing and Correcting for the Effect of Sensor Noise in the Dynamic Mode Decomposition. *Experiments in Fluids*, 57(3):42, Feb. 2016. ISSN 1432-1114. doi: 10.1007/s00348-016-2127-7.

M. Derezinski, D. Calandriello, and M. Valko. Exact sampling of determinantal point processes with sublinear time preprocessing. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

M. Dereziński, M. K. Warmuth, and D. Hsu. Unbiased Estimators for Random Design Regression. *Journal of Machine Learning Research*, 23(167):1–46, 2022. ISSN 1533-7928.

F.-X. L. Dimet and O. Talagrand. Variational Algorithms for Analysis and Assimilation of Meteorological Observations: Theoretical Aspects. *Tellus A*, 38A(2):97–110, 1986. ISSN 1600-0870. doi: 10.1111/j.1600-0870.1986.tb00459.x.

P. Drineas and M. W. Mahoney. On the Nyst\"rom Method for Approximating a Gram Matrix for Improved Kernel-Based Learning. *Journal of Machine Learning Research*, 6(72):2153–2175, 2005. ISSN 1533-7928.

M. F. Driscoll. The reproducing kernel Hilbert space structure of the sample paths of a Gaussian process. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 26(4):309–316, Dec. 1973. ISSN 1432-2064. doi: 10.1007/BF00534894.

C. Dupuy and F. Bach. Learning Determinantal Point Processes in Sublinear Time. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pages 244–257. PMLR, Mar. 2018.

G. Evensen. The Ensemble Kalman Filter: Theoretical Formulation and Practical Implementation. *Ocean Dynamics*, 53(4):343–367, Nov. 2003. ISSN 1616-7341, 1616-7228. doi: 10.1007/s10236-003-0036-9.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, July 2008. ISSN 1465-4644, 1468-4357. doi: 10.1093/biostatistics/kxm045.

M. Gartrell, U. Paquet, and N. Koenigstein. Bayesian Low-Rank Determinantal Point Processes. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 349–356, Boston Massachusetts USA, Sept. 2016. ACM. ISBN 978-1-4503-4035-9. doi: 10.1145/2959100.2959178.

M. Gartrell, U. Paquet, and N. Koenigstein. Low-Rank Factorization of Determinantal Point Processes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v31i1.10869.

G. Gautier, G. Polito, R. Bardenet, and M. Valko. DPPy: DPP Sampling with Python. *Journal of Machine Learning Research*, 20(180):1–7, 2019. ISSN 1533-7928.

J. A. Gillenwater, A. Kulesza, E. Fox, and B. Taskar. Expectation-Maximization for Learning Determinantal Point Processes. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

N. Gordon, D. Salmond, and A. Smith. Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. *IEE Proceedings F Radar and Signal Processing*, 140(2): 107, 1993. ISSN 0956375X. doi: 10.1049/ip-f-2.1993.0015.

A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring Statistical Dependence with Hilbert-Schmidt Norms. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, S. Jain, H. U. Simon, and E. Tomita, editors, *Algorithmic Learning Theory*, volume 3734, pages 63–77, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-29242-5 978-3-540-31696-1. doi: 10.1007/11564089_7.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. ISSN 1533-7928.

M. Gönen and E. Alpaydin. Multiple Kernel Learning Algorithms. *Journal of Machine Learning Research*, 12(64):2211–2268, 2011. ISSN 1533-7928.

M. D. Hoffman and A. Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(47): 1593–1623, 2014. ISSN 1533-7928.

J. Hough, M. Krishnapur, Y. Peres, and B. Virág. *Zeros of Gaussian Analytic Functions and Determinantal Point Processes*, volume 51 of *University Lecture Series*. American Mathematical Society, Providence, Rhode Island, Oct. 2009. ISBN 978-0-8218-4373-4 978-1-4704-1646-1. doi: 10.1090/ulect/051.

J. B. Hough, M. Krishnapur, Y. Peres, and B. Virág. Determinantal Processes and Independence. *Probability Surveys*, 3(none), Jan. 2006. ISSN 1549-5787. doi: 10.1214/154957806000000078.

D. R. Hunter and K. Lange. A tutorial on MM algorithms. *The American Statistician*, 58 (1):30–37, Feb. 2004. ISSN 0003-1305, 1537-2731. doi: 10.1198/0003130042836.

P. Héas and C. Herzet. Low-Rank Dynamic Mode Decomposition: Optimal Solution in Polynomial-Time. *arXiv:1610.02962 [cs, stat]*, Feb. 2020.

M. R. Jovanović, P. J. Schmid, and J. W. Nichols. Sparsity-Promoting Dynamic Mode Decomposition. *Physics of Fluids*, 26(2):024103, Feb. 2014. ISSN 1070-6631, 1089-7666. doi: 10.1063/1.4863670.

R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur. Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences, July 2018.

G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-Informed Machine Learning. *Nature Reviews Physics*, 3(6):422–440, June 2021. ISSN 2522-5820. doi: 10.1038/s42254-021-00314-5.

T. Katō. *Perturbation Theory for Linear Operators*. Classics in Mathematics. Springer, Berlin, 1995. ISBN 978-3-540-58661-6.

Y. Kawahara. Dynamic Mode Decomposition with Reproducing Kernels for Koopman Spectral Analysis. In *Advances in Neural Information Processing Systems 29*, pages 911–919. Curran Associates, Inc., 2016.

T. Kawashima and H. Hino. Gaussian Process Koopman Mode Decomposition. *Neural Computation*, 35(1):82–103, Dec. 2022. ISSN 0899-7667, 1530-888X. doi: 10.1162/neco_a_01555.

T. Kawashima and H. Hino. Minorization-Maximization for Learning Determinantal Point Processes. *Transactions on Machine Learning Research*, May 2023. ISSN 2835-8856.

T. Kawashima, H. Shouno, and H. Hino. Bayesian Dynamic Mode Decomposition with Variational Matrix Factorization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):8083–8091, May 2021. ISSN 2374-3468.

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*. arXiv, 2015. doi: 10.48550/arXiv.1412.6980.

G. Kitagawa. Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996. ISSN 1061-8600. doi: 10.2307/1390750.

V. Kostic, P. Novelli, A. Maurer, C. Ciliberto, L. Rosasco, and M. Pontil. Learning Dynamical Systems via Koopman Operator Regression in Reproducing Kernel Hilbert Spaces. *Advances in Neural Information Processing Systems*, 35:4017–4031, Dec. 2022.

A. Kulesza. *Learning with Determinantal Point Processes*. PhD thesis, University of Pennsylvania, 2012.

A. Kulesza and B. Taskar. K-Dpps: Fixed-Size Determinantal Point Processes. In *International Conference on Machine Learning*, June 2011.

A. Kulesza and B. Taskar. Determinantal Point Processes for Machine Learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286, 2012. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000044.

K. Lange, J.-H. Won, A. Landeros, and H. Zhou. Nonconvex Optimization via MM Algorithms: Convergence Theory. In *Wiley StatsRef: Statistics Reference Online*, pages 1–22. John Wiley & Sons, Ltd, 2021. ISBN 978-1-118-44511-2. doi: 10.1002/9781118445112.stat08295.

G. Last and M. Penrose. *Lectures on the Poisson Process*. Cambridge University Press, 1 edition, Oct. 2017. ISBN 978-1-107-08801-6 978-1-316-10447-7 978-1-107-45843-7. doi: 10.1017/9781316104477.

A. Laub. A Schur Method for Solving Algebraic Riccati Equations. *IEEE Transactions on Automatic Control*, 24(6):913–921, Dec. 1979. ISSN 1558-2523. doi: 10.1109/TAC.1979.1102178.

N. Lawrence. Probabilistic Non-Linear Principal Component Analysis with Gaussian Process Latent Variable Models. *Journal of Machine Learning Research*, 6(60):1783–1816, 2005. ISSN 1533-7928.

S. Le Clainche and J. M. Vega. Higher Order Dynamic Mode Decomposition. *SIAM Journal on Applied Dynamical Systems*, 16(2):882–925, Jan. 2017. doi: 10.1137/15M1054924.

D. Lewandowski, D. Kurowicka, and H. Joe. Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis*, 100(9):1989–2001, Oct. 2009. ISSN 0047259X. doi: 10.1016/j.jmva.2009.04.008.

J. M. Lewis and J. C. Derber. The Use of Adjoint Equations to Solve a Variational Adjustment Problem with Advective Constraints. *Tellus A*, 37A(4):309–322, 1985. ISSN 1600-0870. doi: 10.1111/j.1600-0870.1985.tb00430.x.

C. Li, S. Sra, and S. Jegelka. Fast Mixing Markov Chains for Strongly Rayleigh Measures, DPPs, and Constrained Sampling. In *Advances in Neural Information Processing Systems*, Aug. 2016.

Y. Lian and C. N. Jones. On Gaussian Process Based Koopman Operators. *IFAC-PapersOnLine*, 53(2):449–455, Jan. 2020. ISSN 2405-8963. doi: 10.1016/j.ifacol.2020.12.217.

Y. J. Lim and Y. W. Teh. Variational Bayesian Approach to Movie Rating Prediction. *Proceedings of KDD cup and workshop*, 7:15–21, 2007.

Q. Liu and D. Wang. Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm. *Advances in Neural Information Processing Systems 29*, pages 2378–2386, 2016.

L. Lu, M. Dao, P. Kumar, U. Ramamurty, G. E. Karniadakis, and S. Suresh. Extraction of mechanical properties of materials through deep learning from instrumented indentation. *Proceedings of the National Academy of Sciences*, 117(13):7052–7062, Mar. 2020. doi: 10.1073/pnas.1922210117.

O. Macchi. The Coincidence Approach to Stochastic Point Processes. *Advances in Applied Probability*, 7(1):83–122, 1975. ISSN 0001-8678. doi: 10.2307/1425855.

Z. Mariet and S. Sra. Fixed-Point Algorithms for Learning Determinantal Point Processes. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2389–2397. PMLR, June 2015.

Z. Mariet, M. Gartrell, and S. Sra. Learning Determinantal Point Processes by Corrective Negative Sampling. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pages 2251–2260. PMLR, Apr. 2019a.

Z. E. Mariet. *Learning with Generalized Negative Dependence : Probabilistic Models of Diversity for Machine Learning*. Thesis, Massachusetts Institute of Technology, 2019.

Z. E. Mariet and S. Sra. Kronecker Determinantal Point Processes. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Z. E. Mariet, Y. Ovadia, and J. Snoek. DppNet: Approximating Determinantal Point Processes with Deep Networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019b.

A. Masuda, Y. Susuki, M. Martínez-Ramón, A. Mammoli, and A. Ishigame. Application of Gaussian Process Regression to Koopman Mode Decomposition for Noisy Dynamic Data. *arXiv:1911.01143 [cs, eess, math]*, Nov. 2019.

A. Mathews, M. Francisquez, J. W. Hughes, D. R. Hatch, B. Zhu, and B. N. Rogers. Uncovering turbulent plasma dynamics via deep learning from partial observations. *Physical Review E*, 104(2):025205, Aug. 2021. doi: 10.1103/PhysRevE.104.025205.

I. Mezić. Spectral Properties of Dynamical Systems, Model Reduction and Decompositions. *Nonlinear Dynamics*, 41(1):309–325, Aug. 2005. ISSN 1573-269X. doi: 10.1007/s11071-005-2824-x.

N. Miyoshi and T. Shirai. A Cellular Network Model with Ginibre Configured Base Stations. *Advances in Applied Probability*, 46(3):832–845, Sept. 2014. ISSN 0001-8678, 1475-6064. doi: 10.1239/aap/1409319562.

S. Nakajima and M. Sugiyama. Theoretical Analysis of Bayesian Matrix Factorization. *Journal of Machine Learning Research*, 12(79):2583–2648, 2011. ISSN 1533-7928.

A. Nemirovsky. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience Series in Discrete Mathematics. New York: Wiley, 1983. ISBN 978-0-471-10345-5.

T. Osogami, R. Raymond, A. Goel, T. Shirai, and T. Maehara. Dynamic Determinantal Point Processes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. ISSN 2374-3468. doi: 10.1609/aaai.v32i1.11598.

N. Parikh and S. Boyd. Proximal Algorithms. *Found. Trends Optim.*, 1(3):127–239, Jan. 2014. ISSN 2167-3888. doi: 10.1561/2400000003.

J. L. Proctor and P. A. Eckhoff. Discovering Dynamic Patterns from Infectious Disease Data Using Dynamic Mode Decomposition. *International Health*, 7(2):139–145, Mar. 2015. ISSN 1876-3413. doi: 10.1093/inthealth/ihv009.

A. Rahimi and B. Recht. Random Features for Large-Scale Kernel Machines. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

B. Rakitsch, C. Lippert, K. Borgwardt, and O. Stegle. It Is All in the Noise: Efficient Multi-Task Gaussian Process Inference with Structured Residuals. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Mass., 3. print edition, 2008. ISBN 978-0-262-18253-9.

C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson. Spectral Analysis of Nonlinear Flows. *Journal of Fluid Mechanics*, 641:115–127, Dec. 2009. ISSN 0022-1120, 1469-7645. doi: 10.1017/S0022112009992059.

S. Saitoh and Y. Sawano. *Theory of Reproducing Kernels and Applications*, volume 44 of *Developments in Mathematics*. Springer, Singapore, 2016. ISBN 978-981-10-0529-9 978-981-10-0530-5. doi: 10.1007/978-981-10-0530-5.

Y.-L. K. Samo and S. Roberts. Scalable Nonparametric Bayesian Inference on Point Processes with Gaussian Processes. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2227–2236. PMLR, June 2015.

P. J. Schmid. Dynamic Mode Decomposition of Numerical and Experimental Data. *Journal of Fluid Mechanics*, 656:5–28, Aug. 2010. ISSN 0022-1120, 1469-7645. doi: 10.1017/S0022112010001217.

B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In G. Goos, J. Hartmanis, J. Van Leeuwen, W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, editors, *Artificial Neural Networks — ICANN'97*, volume 1327, pages 583–588, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg. ISBN 978-3-540-63631-1 978-3-540-69620-9. doi: 10.1007/BFb0020217.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings* *of the IEEE*, 104(1):148–175, Jan. 2016. ISSN 0018-9219, 1558-2256. doi: 10.1109/JPROC.2015.2494218.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, 2004. doi: 10.1017/CBO9780511809682.

T. Shirai and Y. Takahashi. Fermion Process and Fredholm Determinant. In *Proceedings of the Second ISAAC Congress*, pages 15–23. Springer, 2000.

V. Simoncini. Computational Methods for Linear Matrix Equations. *SIAM Review*, 58 (3):377–441, Jan. 2016. ISSN 0036-1445, 1095-7200. doi: 10.1137/130912839.

A. Soshnikov. Determinantal Random Point Fields. *Russian Mathematical Surveys*, 55(5):923–975, Oct. 2000. ISSN 0036-0279, 1468-4829. doi: 10.1070/RM2000v055n05ABEH000321.

O. Stegle, C. Lippert, J. M. Mooij, N. Lawrence, and K. Borgwardt. Efficient Inference in Matrix-Variate Gaussian Models with \iid Observation Noise. *Advances in Neural Information Processing Systems*, 24, 2011.

Y. Sun, P. Babu, and D. P. Palomar. Robust Estimation of Structured Covariance Matrix for Heavy-Tailed Elliptical Distributions. *IEEE Transactions on Signal Processing*, 64 (14):3576–3590, July 2016. ISSN 1941-0476. doi: 10.1109/TSP.2016.2546222.

Y. Sun, P. Babu, and D. P. Palomar. Majorization-Minimization Algorithms in Signal Processing, Communications, and Machine Learning. *IEEE Transactions on Signal Processing*, 65(3):794–816, Feb. 2017. ISSN 1053-587X, 1941-0476. doi: 10.1109/TSP.2016.2601299.

N. Takeishi, Y. Kawahara, Y. Tabei, and T. Yairi. Bayesian Dynamic Mode Decomposition. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 2814–2821, Melbourne, Australia, Aug. 2017a. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-0-3. doi: 10.24963/ijcai.2017/392.

N. Takeishi, Y. Kawahara, and T. Yairi. Learning Koopman Invariant Subspaces for Dynamic Mode Decomposition. *Advances in Neural Information Processing Systems*, 30, 2017b.

M. Titsias. Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *Artificial Intelligence and Statistics*, pages 567–574. PMLR, Apr. 2009.

M. Titsias and N. D. Lawrence. Bayesian Gaussian Process Latent Variable Model. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 844–851. JMLR Workshop and Conference Proceedings, Mar. 2010.

D. Vere-Jones. Alpha-permanents and their applications to multivariate gamma, negative binomial and ordinary binomial distributions. *New Zealand J. Math*, 26(1):125–149, 1997.

J. Wang, A. Hertzmann, and D. J. Fleet. Gaussian Process Dynamical Models. *Advances in Neural Information Processing Systems*, 18, 2005.

J. Wang, D. Fleet, and A. Hertzmann. Gaussian Process Dynamical Models for Human Motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2): 283–298, Feb. 2008. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.1167.

M. Welling. The Kalman Filter. *Lecture Note*, pages 92–117, 2010.

V. Wild, M. Kanagawa, and D. Sejdinovic. Connections and Equivalences Between the Nystr\"om Method and Sparse Variational Gaussian Processes. *arXiv:2106.01121 [cs, math, stat]*, June 2021.

C. Williams and M. Seeger. Using the Nyström Method to Speed Up Kernel Machines. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000.

M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A Data–Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, Dec. 2015a. ISSN 1432-1467. doi: 10.1007/s00332-015-9258-5.

M. O. Williams, C. W. Rowley, and I. G. Kevrekidis. A Kernel-Based Method for Data-Driven Koopman Spectral Analysis. *Journal of Computational Dynamics*, 2(2): 247–265, 2015b. doi: 10.3934/jcd.2015005.

World Health Organization. *Evolution of a Pandemic: A(H1N1) 2009, April 2009 – August 2010*. World Health Organization, Geneva, 2nd ed. edition, 2013. ISBN 978-92-4-150305-1.

A. L. Yuille and A. Rangarajan. The Concave-Convex Procedure (CCCP). In *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001.

W. Zhu, K. Xu, E. Darve, and G. C. Beroza. A general approach to seismic inversion with automatic differentiation. *Computers & Geosciences*, 151:104751, June 2021. ISSN 0098-3004. doi: 10.1016/j.cageo.2021.104751.

N. Črnjarić-Žic, S. Maćešić, and I. Mezić. Koopman Operator Spectrum for Random Dynamical Systems. *Journal of Nonlinear Science*, 30(5):2007–2056, Oct. 2020. ISSN 1432-1467. doi: 10.1007/s00332-019-09582-z.