LOCALLY STATIONARY GRAPH PROCESSES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ABDULLAH CANBOLAT

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2023

Approval of the thesis:

LOCALLY STATIONARY GRAPH PROCESSES

submitted by **ABDULLAH CANBOLAT** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** _____

Assoc. Prof. Dr. Elif Vural
Supervisor, **Electrical and Electronics Engineering, METU** _____

Examining Committee Members:

Assist. Prof. Dr. Barış Nakiboğlu
Electrical and Electronics Engineering, METU _____

Assoc. Prof. Dr. Elif Vural
Electrical and Electronics Engineering, METU _____

Assoc. Prof. Dr. Cem Tekin
Electrical and Electronics Engineering, Bilkent University _____

Date: 20.12.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Abdullah Canbolat

Signature :

ABSTRACT

LOCALLY STATIONARY GRAPH PROCESSES

Canbolat, Abdullah
M.S., Department of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Elif Vural

December 2023, 81 pages

Stationary graph process models are commonly used in the analysis and inference of data sets collected on irregular network topologies. While most of the existing methods represent graph signals with a single stationary process model that is globally valid on the entire graph, in many practical problems, the characteristics of the process may be subject to local variations in different regions of the graph. In this thesis, we propose a locally stationary graph process (LSGP) model that aims to extend the classical concept of local stationarity to irregular graph domains. We characterize local stationarity by expressing the overall process as the combination of a set of component processes such that the extent to which the process adheres to each component varies smoothly over the graph. We propose an algorithm for computing LSGP models from realizations of the process, and also study the approximation of LSGPs locally with Wide Sense Stationary (WSS) processes. Experiments on signal interpolation problems show that the proposed process model provides accurate signal representations competitive with the state of the art.

Keywords: Locally Stationary Graph Processes, Non-stationary Graph Processes,

Graph Signal Interpolation, Graph Partitioning

ÖZ

YEREL DURAĞAN ÇİZGE SÜREÇLERİ

Canbolat, Abdullah
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Elif Vural

Aralık 2023 , 81 sayfa

Durağan çizge süreç modelleri, düzensiz ağ topolojilerinde toplanan veri kümelerinin analiz ve çıkarımında yaygın olarak kullanılmaktadır. Mevcut yöntemlerin çoğu tüm çizgenin tamamında geçerli olan tek bir durağan süreç modeli ile çizge işaretlerini temsil ederken, birçok pratik problemde sürecin özellikleri çizgenin farklı bölgelerinde yerel değişikliklere tabi olabilir. Bu tezde klasik yerel durağanlık kavramını düzensiz çizge topolojilerine genişletmeyi amaçlayan yerel olarak durağan bir çizge süreç (LSGP) modeli önerilmiştir. Geliştirilen modelde çizge süreci bir dizi bileşen sürecin kombinasyonu olarak ifade edilmiş, yerel durağanlık özelliği ise her bir bileşen sürecin modeldeki ağırlığının çizge üzerinde yavaş bir şekilde değişmesi ile karakterize edilmiştir. Sürecin örnekleminden LSGP modellerini hesaplamak için bir algoritma önerilmiş ve ayrıca LSGPlerin yerel olarak geniş anlamda durağan (WSS) süreçler ile yaklaşık olarak gösterimi incelenmiştir. İşaret tamamlama problemleri üzerine yapılan deneyler, önerilen süreç modelinin, literatürdeki güncel algoritmalarla rekabet edebilecek doğru işaret temsilleri sağladığını göstermektedir

Anahtar Kelimeler: Yerel Durağan Çizge Süreçleri, Durağan Olmayan Çizge Süreçleri, Çizge İşareti Tamamlama, Çizge Bölümleme

Ad prosperitatem liberi cogitatus...

ACKNOWLEDGEMENTS

I see this thesis as an accumulation of the experiences and knowledge I have gained from numerous people and places. I dedicate this space to appreciate their assistance, support, and love.

I would like to thank my thesis supervisor, Assoc. Prof. Dr. Elif Vural, for her invaluable guidance, understanding, and expertise. This thesis would not have been possible without her help.

I also express my gratitude to Eylem Tuğçe Güneyi and Berkay Yaldız. Our collaborative studies have taught me a great deal, shaping the experimental results that are integral to this thesis.

I am grateful to my current colleagues at METU and my former colleagues at ASELSAN. Although my time with them was brief, it was immensely enjoyable and educational, significantly enhancing my understanding of network theory.

Last but not least, my heartfelt thanks go to my parents and my sisters, Nüveyre and Nurgül, for their unwavering support during the challenging moments of this thesis. I am equally grateful to İrem for her love and support. Their encouragement and aid have been indispensable.

In short, I am grateful to everyone who has supported me, whether they came to mind just now or not.

This thesis study was supported by the Scientific and Technological Resesarch Council of Turkey (TUBITAK) under grant 120E246.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

AR	Auto Regressive
ARMA	Auto Regressive Moving Average
CD	Covariance Discrepancy
FT	Fourier Transform
GFT	Graph Fourier Transform
GSP	Graph Signal Processing
HKM	Helmberg-Kojima-Monteiro
k-NN	k Nearest Neighbors
LMMSE	Linear Minimum Mean Square Estimate
LSGP	Locally Stationary Graph Process
LSP	Locally Stationary Process
MA	Moving Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
NME	Normalized Mean Error
NMI	Normalized Mutual Information
PSD	Positive SemiDefinite
QP	Quadratic Program
SDP	Semidefinite Programming
SNR	Signal-to-Noise Ratio
SOCP	Second Order Cone Program
STFT	Short Time Fourier Transform

SVD	Singular Value Decomposition
WSS	Wide Sense Stationary

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Analysis of acquired data has been a critical focus in signal processing for many years. A classical approach has been used for data analysis since its introduction: Fourier Transform (FT). FT is efficient in a way that it can capture the frequency characteristics of the data, allowing interpretable representations. However, one major drawback of FT is that it assumes the data has an underlying grid structure, which is a standard connection between the instances of the data. This may not always be the case, because even a simple graph topology constructed by connecting diagonals of a 2d grid structure does not meet this assumption. This problem is overcome with emergence of Graph Signal Processing (GSP).

GSP has successfully extended common data analysis tools, such as FT, sampling, dictionary learning and other data analysis approaches [1–3] to irregular graph domains. Moreover, in the literature, the probabilistic modelling of a signal defined over graphs is introduced in several works, and Wide Sense Stationarity (WSS) of graph signals [4–7], is a crucial development in this direction. A WSS model imposes a signal structure that is compatible with the given topology. However, this is rather strict, asserting a global implication on the stochastic model, e.g. in classical time series analysis, a WSS model has a Moving Average (MA) representation on time domain that is valid for all time instants [8].

In the time domain, this problem is overcome with the development of the Locally Stationary Process (LSP) model [9], which assumes that WSS process polynomial has smooth progression as time passes, allowing the decomposition of the domain

into several pieces, hence modeling a time series with several different WSS models that are arbitrarily close to the original process on each interval. A similar notion is used in Piecewise Stationary Processes, which assume that these transitions are abrupt rather than smooth, hence each interval is modelled without considering the smoothness of the polynomial coefficients, and in return, the covariance matrix [10].

Piecewise Stationary Process models are successfully extended to irregular graph topologies [11], however, a rigorous treatment of LSPs on graph domains is missing in the literature. In this thesis, we aim to generalize the notion of LSP to graph domains with a rigorous theoretical treatment.

1.2 Proposed Methods and Models

In this thesis, we provide a definition for a locally stationary graph process (LSGP) model, which can be thought as consisting of different WSS kernels smoothly transitioning from one vertex to another. The transition and validity of a given WSS process are controlled via parameters called membership functions. This will help us to apply the discussion in LSP conveniently in graph domains. First, this provides us with a clear notion of smoothness of the process, second, it increases the explainability of the model. Moreover, thanks to this explainability we get a representation called the vertex-frequency spectrum. Since any LSGP has a vertex-frequency spectrum we see that WSS processes are subsets of this general class if we take any kernel that does not change through the graph. Furthermore, this property of the process will lead to the consequence that the space of LSGPs on subgraphs of the process can be embedded on the space of LSGPs on their supergraphs through the process extension, which might be useful for specific applications concerning the subgraph of a specific network. Its inverse process called the restriction is also another applicable operation on these processes, hence going back to the subgraph process is also possible, of course with an expected data loss.

Following the definition of the process, a possible question to ask is how one can learn this process. We address this question by introducing a learning algorithm consisting of a nonconvex optimization problem utilizing the sample covariance matrix of the

process. Since this nonconvex problem is intractable, we solve it by introducing positive semidefinite (PSD) matrices that are second-order functions of the process parameters, and using an alternating optimization scheme. This allows us to learn the parameters of the process.

The convex relaxation of the learning algorithm enables an accurate computation of the process parameters; however, this comes at the cost of an increase in the time complexity of the algorithm. Therefore, the subsequent inquiry is whether it is possible to further decrease the complexity. Since we use the covariance matrix of the process and models are assumed to concentrate on some graph regions, it is possible to theoretically study the covariance matrix of a pure LSGP. The main expectation on this issue is that under the condition that the membership functions are concentrated on some subregions of the graph, the cross correlation between those regions should be considerably small. In this thesis, this is proved to be true under the condition that the frequency spectrums and membership functions of the components of the LSGP are clearly distinguishable from each other, hence highlighting the importance of the vertex-frequency spectrum of the process. Assuming that the membership functions and frequency spectrums are clearly separated, we propose a graph partitioning algorithm that approximates the overall LSGP via its components defined on the subgraphs, where we utilize the covariance values on the edges of the graph.

1.3 Contributions and Novelties

This thesis study presents some comprehensive contributions over the existing literature, which can be summarized as follows:

- We propose a novel signal model called LSGP and study the theoretical properties of this model. To the best of our knowledge, this model has not been explored or defined thoroughly in the GSP literature until this work. Moreover, we introduce an algorithm for learning the parameters of an LSGP model from an initial estimate of the covariance matrix of the process.
- We theoretically study the properties of the covariance matrix of an LSGP, and through this analysis, propose a graph partioning algorithm utilizing the covari-

ance matrix. The idea of using process values for graph partitioning is not new, e.g. there is an example of it in [11], but our partitioning algorithm uses the covariance matrix of the process.

- We demonstrate the usage of the proposed model in signal interpolation applications. We estimate the missing entries of graph signals with the proposed process model, using the available observations to estimate the covariance matrix, in contrary to algorithms requiring the whole process for estimating a model.

1.4 The Outline of the Thesis

In Chapter 2, we introduce the related literature, which overviews some basic concepts for understanding locally stationary processes. First, an outline of graph signal processing and spectral graph theory will be given. Second, we discuss stochastic processes defined over some topology and its extension to graphs. We will then explore two definitions of LSPs found in the literature, giving an idea of our model, i.e. how it would behave on irregular topologies. Also, we review examples of similar approaches for modelling graph processes in the literature. Last but not least, we conclude the Chapter with a literature review on the estimation of missing signals.

In Chapter 3, we introduce our LSGP model and investigate its theoretical properties, vertex-frequency spectrum and extension-restriction of the process. We show that our process model is comprehensive enough to model the process on the subgraphs separately.

In Chapter 4, we present an algorithm to learn process parameters for a given data set and propose an optimization algorithm. Then, the problem in its explicit final form is given. Lastly, we present a complexity analysis of the algorithm.

In Chapter 5, we discuss theoretical properties of covariance matrices of LSGPs in order to address the high complexity of the algorithm in Chapter 4. Then, we give a partitioning algorithm on the graph utilizing the covariance information of the given process.

In Chapter 6, we present the experimental results of the algorithms proposed in Chap-

ters 4 and 5 and elaborate further on the practical properties of LSGPs.

In Chapter 7, we conclude the thesis by highlighting possible future extensions.

CHAPTER 2

RELATED WORK

In this chapter, we will revisit some key concepts in spectral graph theory and signal processing in both the graph and time domains by reviewing recent advances in the literature.

2.1 Notation

We write matrices with boldface capital letters (e.g. $\mathbf{A}$), vectors with boldface lowercase letters (e.g. $\mathbf{x}$) and sets with caligraphic letters (e.g. $\mathcal{V}$). Indexing is shown in parentheses (e.g. $\mathbf{A}(i,j)$). The notation $\|\cdot\|_F$ stands for the Frobenius norm of a matrix; $\circ$ refers to the Hadamard (elementwise) product, $(\cdot)^T$ denotes the transpose and $(\cdot)^\dagger$ denotes the pseudo-inverse of a matrix. Vectors and matrices consisting of only zeros and only ones are denoted as $\mathbf{0}_N \in \mathbb{R}^{N\times 1}$, $\mathbf{1}_N \in \mathbb{R}^{N\times 1}$ and $\mathbf{0}_{N\times M} \in \mathbb{R}^{N\times M}$, $\mathbf{1}_{N\times M} \in \mathbb{R}^{N\times M}$, respectively. $\mathbf{I}_N \in \mathbb{R}^{N\times N}$ represents the identity matrix. When written as a matrix operation, the notation $|\cdot| : \mathbb{R}^{N\times M} \to \mathbb{R}^{N\times M}$ stands for element-wise absolute value, i.e., $|\mathbf{A}|(i,j) = |\mathbf{A}(i,j)|$ for a matrix $\mathbf{A}$. Curly inequalities for matrices represent element-wise inequality, e.g. $\mathbf{A} \preccurlyeq \mathbf{B}$ means $\mathbf{A}(i,j) \leq \mathbf{B}(i,j)$ for all entries (i,j). The cardinality of a set is denoted as $|\cdot|$. The covariance matrix of a random vector $\mathbf{x}$ is shown as $\mathbf{C}_\mathbf{x}$.

2.2 Graph Signal Processing

In this thesis, we consider graphs defined with tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$. $\mathcal{V}$ is the vertex set whose entries are called vertices of the graph $\mathcal{G}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set

of the graph. The entries of $\mathcal{E}$ are referred to as edges, which illustrate the 2-tuple dependencies between the graph's vertices. Moreover, we impose that any element in the edge set is invariant to the permutation of its corresponding entries, i.e. if $(v, w) \in \mathcal{E}$ then $(w, v) \in \mathcal{E}$. Such graphs are called undirected graphs. For an undirected graph the edge relation can also be shown with the notation $v \sim w$, if $(v, w) \in \mathcal{E}$. If all the edges are not invariant to permutation, then the graph is called directed. Throughout the thesis, we assume that $\mathcal{G}$ is undirected. The reason why we are using undirected graphs instead of directed ones is that undirected graphs have well-defined spectral properties allowing us to generalize the notion of frequency in a straightforward way, contrary to directed graphs [1]. However, there is some literature extending the idea of frequency to directed graphs as well with some tricks [12–14]. We furthermore enrich the structure on the graph $\mathcal{G}$ with $\mathbf{W} : \mathcal{E} \to \mathbb{R}_{\geq 0}$ called weights of the graph $\mathcal{G}$. Let $N = |\mathcal{V}|$, then one can represent this edge function with the matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ with an abuse of notation. A precise definition of $\mathbf{W}$ can be given by picking some enumeration of the vertices. The enumeration of the vertices can be given in terms of an arbitrary bijective function $f : \mathcal{V} \to \{1, 2, \ldots, N\}$, relating a natural number to each vertex. Again with some abuse of notation, it can be defined as

$$\mathbf{W}(i, j) = \begin{cases} 0 & (f^{-1}(i), f^{-1}(j)) \notin \mathcal{E} \\ \mathbf{W}(f^{-1}(i), f^{-1}(j)) & \text{otherwise} \end{cases} \tag{2.1}$$

In graph signal processing, we mainly focus on signals on a point cloud scattered in $\mathbb{R}^N$. Among several approaches in the literature [15–17], we will use the k-Nearest Neighbors (k-NN) algorithm to construct the underlying graph with Gaussian edge weights in (2.2), where σ is a positive hyperparameter that controls the scattering of the weights and $\mathbf{x}_i$ is the ith element in the point cloud.

$$\mathbf{W}(v, w) = \exp\left(-\frac{\|\mathbf{x}_{f(v)} - \mathbf{x}_{f(w)}\|^2}{\sigma}\right) \tag{2.2}$$

A graph $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s, \mathbf{W}_s)$ is said to be a proper subgraph (subgraph in short) of $\mathcal{G}$ if $\mathcal{V}_s \subset \mathcal{V}$ and $\mathcal{E}_s \subseteq (\mathcal{V}_s \times \mathcal{V}_s) \cap \mathcal{E}$ and $\mathbf{W}_s = \mathbf{W}|_{\mathcal{E}_s}$. A graph $\mathcal{G}$ is said to be a supergraph

of $\mathcal{G}_s$ if $\mathcal{G}_s$ is a subgraph of $\mathcal{G}$.

Before delving into details on spectral graph theory, let us give the definition of a connected graph. A graph is said to have a single connected component if for each pair of vertices v, w, there exists some sequence of vertices $\{v, p_1, \ldots, p_{n-1}, w\}$ such that $\forall i, (v, p_1), (p_i, p_{i+1})$ and $(p_{n-1}, w) \in \mathcal{E}$. Graphs with multiple connected components are formed by taking unions of single connected components. All graphs in this thesis are assumed to have single connected components.

For spectral analysis on the graphs, we assume some structure matrix $\mathbf{S}$ based on the information encoded with $\mathbf{W}$. There are several choices of $\mathbf{S}$ considered in the literature [5, 18, 19], and one of the common choices is to use the Laplacian matrix $\mathbf{L}_{\mathcal{G}}$ of the graph [6,20,21], which provides the generalization of the second derivative to irregular discrete spaces. The definition of the classical Laplacian matrix is based on the degree of each vertex. The degree of a vertex is defined in (2.3).

$$d_v = \sum_{w \sim v} \mathbf{W}(v, w) \tag{2.3}$$

Moreover, vertex degrees can be embedded into a diagonal matrix $\mathbf{D}$ with the same enumeration $\mathbf{D}(f(v), f(v)) = d_v$. Then, the classical Laplacian matrix is defined as in (2.4).

$$\hat{\mathbf{L}_{\mathcal{G}}} = \mathbf{D} - \mathbf{W} \tag{2.4}$$

Since Laplacian is the generalization of the second derivative on graphs, one can define the notion of frequency on graphs, similarly to the case where the eigenvalues of the second derivative operator on some closed interval with Dirichlet boundary conditions gives the square valued frequencies of Fourier Series and eigenvectors are sine and cosine functions [22]. By construction, $\hat{\mathbf{L}_{\mathcal{G}}}$ is a positive semidefinite symmetric matrix whose eigenvalue decomposition can be given as in (2.5).

$$\hat{\mathbf{L}_{\mathcal{G}}} = \hat{\mathbf{U}_{\mathcal{G}}} \hat{\mathbf{\Lambda}_{\mathcal{G}}} \hat{\mathbf{U}_{\mathcal{G}}}^T \tag{2.5}$$

Due to its construction, the classical Laplacian matrix always has its lowest eigenvalue as 0 which corresponds to the constant vector [1]. The multiplicity of the 0 eigenvalue corresponds to the number of connected components [23], thus, in our case we observe the 0 eigenvalue with multiplicity 1.

One downside of this construction is that for increasing number of vertices the maximum eigenvalue of the graph Laplacian will increase accordingly. As we will see in Section 3.1.2.1, any LSGP can be extended to a supergraph with regular arguments, hence allowing a graph to grow indefinitely means its maximum eigenvalue will also increase unboundedly hence keeping track of the process may not be possible. Moreover, this may be an issue for numerical computations, making them impossible to implement practically. These problems will be overcome with the normalized Laplacian definition in (2.6a) with the decompositon in (2.6b). In this case, the eigenvalues of $\mathbf{L}_{\mathcal{G}}$ are scattered in between $[0, 2]$, no matter what size the graph is [23].

$$\mathbf{L}_{\mathcal{G}} = \mathbf{I}_{N\times N} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} = \mathbf{D}^{-1/2}\hat{\mathbf{L}_{\mathcal{G}}}\mathbf{D}^{-1/2} \tag{2.6a}$$

$$\mathbf{L}_{\mathcal{G}} = \mathbf{U}_{\mathcal{G}}\mathbf{\Lambda}_{\mathcal{G}}\mathbf{U}_{\mathcal{G}}^{T} \tag{2.6b}$$

Once we have completed the review of spectral graph theory, we will now focus on the signals defined over the vertices of the graph. The spectral properties of the graph is useful for the analysis of the graph signals. A graph signal $\mathbf{x}$ can be defined as a function between the vertex set and a vector space. For example, if vertices are mapped to a real number then the vector space is $\mathbb{R}$, if vertices are mapped to a random variable then the vector space is the space of random variables. More rigorously, $\mathbf{x} : \mathcal{V} \to \mathbb{R}$, considering that the vector space is real numbers. Considering the enumeration f, $\mathbf{x}$ can be represented by a vector $\mathbf{x} \in \mathbb{R}^N$ as $\mathbf{x}(f(v)) = \mathbf{x}(v)$ with an abuse of notation. From this point and on, since the operators discussed here are enumeration invariant, we will drop the enumeration f and show the resulting vector with vertex entries.

As aformentioned since the Laplacian matrix is the generalization of the second derivative, its eigenvectors can be seen as the extension of Fourier Series to graphs. Hence one may define the Graph Fourier Transform (GFT) as the application of trans-

posed eigenvectors to the signal $\mathbf{x}$ in (2.7) and similarly define its inverse.

$$\hat{\mathbf{x}} = \mathbf{U}_{\mathcal{G}}^{T}\mathbf{x} \qquad \text{(GFT)} \tag{2.7a}$$

$$\mathbf{x} = \mathbf{U}_{\mathcal{G}}\hat{\mathbf{x}} \qquad \text{(Inverse GFT)} \tag{2.7b}$$

Similarly to the case in classical signal processing, filtering is done via applying a kernel to the eigenvalues of the graph Laplacian, i.e. the graph frequencies. A simple filtering operation can be viewed in (2.8a) with a frequency kernel $h : \mathbb{R} \to \mathbb{R}$ and for a graph signal $\mathbf{x}$. A compact form of the same operation where the kernel is a matrix function applied to the Laplacian can be viewed in (2.8b).

$$\mathbf{y} = \mathbf{U}_{\mathcal{G}} h(\mathbf{\Lambda}_{\mathcal{G}})\mathbf{U}_{\mathcal{G}}^{T}\mathbf{x} \tag{2.8a}$$

$$\mathbf{y} = h(\mathbf{L}_{\mathcal{G}})\mathbf{x} \tag{2.8b}$$

The representation in (2.8a) is important in the analysis of LSGPs because with some trick, it can be changed into a form where one can see how much each node is affected from the filtering operation. This form is shown in (2.9). More details about this form can be found in Section 3.1.1. Note that for frequency filtering, each node is filtered with the same frequency kernel which is encoded in a vector of ones.

$$\mathbf{y} = (\mathbf{U}_{\mathcal{G}} \circ \mathbf{1}h(\lambda_{\mathcal{G}})^{T})\mathbf{U}_{\mathcal{G}}^{T}\mathbf{x} \tag{2.9}$$

It's worth noting that if the graph topology is kept constant and all eigenvalues have multiplicity 1, any kernel h can be written as a polynomial with highest order $N-1$ due to Cayley-Hamilton theorem [24]. This fact will be useful for the analysis of WSS processes while stating that any WSS process can be written as a MA process.

In classical signal processing, it is desirable to have signals changing smoothly in the domain. One prominent example can be given as the Nyquist Sampling Theorem, which requires the sampled signal to be smooth enough, i.e. to have low-pass behaviour. Such signals are reconstructable from their discretized values sampled at

evenly spaced intervals [25]. Hence, it is customary to use the frequency characteristics of the signal to see how smooth the signal is. But this is rather restrictive in GSP because GFT has high complexity $\mathrm{O}(N^3)$ contrary to the complexity of Fast FT, $\mathrm{O}(N\log_2(N))$ [26]. There are several works addressing this issue, which decrease the complexity of the GFT operation for special cases of graphs [27,28]. Therefore, most of the time in GSP, the smoothness of a graph signal $\mathbf{x}$ is measured by (2.10) [1]. In the common situation where the graph Laplacian is sparse, the complexity of verifying whether the signal exhibits a low-pass behaviour decreases.

$$\text{Smoothness} = \mathbf{x}^T\mathbf{L}_{\mathcal{G}}\mathbf{x} =< \mathbf{L}_{\mathcal{G}}\mathbf{x}, \mathbf{x} > \tag{2.10}$$

(2.10) is higher when the signal is rapidly changing and lower when the signal is smoothly changing between vertices. The term (2.10) can be interpreted as an inner product where high frequency components are penalized more hence detecting the presence of rapidly changing components. In this thesis we use (2.10) to measure the smoothness of the membership functions.

2.3 Wide Sense Stationary Processes

In this section, we will divert our focus to stochastic processes with strong properties both in time and vertex domains. Consider a stochastic process $\mathbf{x}$ whose index set is integers or real numbers. Then $\mathbf{x}$ is called a WSS process on that domain if it satisfies the properties in (2.11). Hence, WSS processes in the time domain is characterized by the invariance of the process characteristics to the shift operator. This allows a spectral representation of the covariance $\mathbf{C}_{\mathbf{x}}$ [29].

$$E[\mathbf{x}(t)] = c, c \in \mathbb{R}, \forall t \tag{2.11a}$$

$$\mathbf{C}_{\mathbf{x}}[t_1, t_2] = E[\mathbf{x}(t_1)\mathbf{x}(t_2)] = \mathbf{C}_{\mathbf{x}}[t_1 - t_2], \forall t_1, t_2 \tag{2.11b}$$

Inspired by these observations WSS processes on graphs are defined as in Definition 1.

Definition 1 (Wide Sense Stationary Process)**.** *A stochastic graph signal* $\mathbf{x} \in \mathbb{R}^N$ *on graph* $\mathcal{G}$ *is called WSS if the following conditions are satisfied [6]:*

- *The mean of the process is* $E[\mathbf{x}] = c\,\mathbf{1}_N$ *for a constant* $c \in \mathbb{R}$*.*
- *The covariance matrix* $\mathbf{C_x}$ *of the process is in the form of a graph filter* $\mathbf{C_x} = \mathbf{U}_{\mathcal{G}} h(\mathbf{\Lambda}_{\mathcal{G}})\mathbf{U}_{\mathcal{G}}^T$*, where* h *is a positive graph kernel.*

The second condition here imposes that the covariance matrix $\mathbf{C_x}$ of the process $\mathbf{x}$ be characterized by a single graph kernel, which is equivalent to the case in the time domain, where the covariance matrix of a WSS process has a spectral decomposition.

Utilizing AR, MA and ARMA filters is a common practice in GSP [5,20,30,31]. Here we will make it clear why it is a valid choice through implications on WSS process in the time domain. According to Wold's Theorem [32] any covariance stationary process in the time domain can be written by a MA filter with infinite support plus a mean term. On the other hand, AR filters may be written as an infinitely supported MA filter, e.g., consider the simple example in signal processing $\sum_{n=0}^{\infty} \alpha^n z^{-n} = \frac{1}{1-\alpha z^{-1}}$ or vice versa. Therefore, AR, MA or ARMA filters are mainly used in signal processing for various purposes. In the case of GSP, each of these filters can be generated by a MA filter due to the reason given in Section 2.2. Moreover, assuming an AR filter in Section 4.1 introduces an irreducible non-convexity to the objective function. Therefore, we mainly focus on MA filters in this thesis.

Since there is a single graph kernel constituting a covariance matrix, as we observed in (2.9), this is a global assumption on a given topology that does not account for the local changes across the graph. Hence we will explore local approaches in the time domain to generalize them to the graphs.

2.4 Locally Stationary Time Processes

A non-stationary process that is locally close to a stationary one has been a concern in the classical time domain setting for many years. In this section, we will explore two different definitions in the time domain by discussing their estimation techniques.

These definitions are due to Silverman [33] and Dahlhaus [9].

According to Silverman a process $\mathbf{x}$ is called LSP if its covariance $\mathbf{C_x}$ can be decomposed as in (2.12), where $\mathbf{C}$ is a WSS covariance and $\mathbf{Q}$ is a function showing the validity of the covariance $\mathbf{C}$ at different time instants [33]. However, this is rather strict because this process is generated by a single WSS covariance function by varying the validity of the covariance at the midpoints of time instant pairs. Since this definition does not allow the incorporation of the effect of more than one covariance function into the process model, it is less flexible than our definition.

$$\mathbf{C_x}(t_1, t_2) = \mathbf{Q}\left(\frac{t_1 + t_2}{2}\right)\mathbf{C}(t_1 - t_2) \tag{2.12}$$

Although (2.12) can be seen as a special case of our definition where the number of models is taken to be 1, some estimation techniques utilizing this definition is worth looking at. In [34], for the special case of LSPs whose process values are i.i.d. under rotation, optimal window and data tapers are given for the estimation of the Wigner-Ville Spectrum [35], a time varying spectrum for covariances in the time domain. In the same work a generalization of Silverman's LSP called Multicomponent LSP is proposed, which might be seen as an equivalent formulation of our definition in the time domain. An optimal filter similar to the Wiener Filter is devised in [36], which is used for EEG application. Another work on estimation approaches can be found in [37], where $\mathbf{Q}$ and $\mathbf{C}$ are found separately, similar to our approach where we find process parameters in an alternating manner (see Chapter 4). One thing to note is that, contrary to these approaches our model consists of several component models in which case the generalization of these approaches may not be feasible.

Let $x_{t,T}$ be a finite-dimensional random process, where $t \in \mathbb{Z}$ is a time instant and $T \in \mathbb{Z}^+$ is the total number of time instants in the process. Also, let ϵ_t represent the value of a white noise process at time instant t. According to Dahlhaus, the process $x_{t,T}$ is then called a linear locally stationary process if it can be expressed as a time-varying MA(∞) process as [9]

$$x_{t,T} = \sum_{i=-\infty}^{\infty} a_{t,T}(i)\,\epsilon_{t-i} + \mu\left(\frac{t}{T}\right). \tag{2.13}$$

Here $\mu : [0,1] \rightarrow \mathbb{R}$ is the mean function of the process and $a_{t,T} : \mathbb{Z} \rightarrow \mathbb{R}$ are time-varying moving average (MA) filter coefficients. For each time shift value i, the filter coefficient $a_{t,T}(i)$ is approximately given through a kernel f as $a_{t,T}(i) \approx f(\frac{t}{T}, i)$ where $f(., i) : [0,1] \rightarrow \mathbb{R}$.

The process $x_{t,T}$ has the spectral decomposition [9]

$$x_{t,T} = \mu\left(\frac{t}{T}\right) + \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} A_{t,T}(\lambda) e^{j\lambda t} d\xi(\lambda) \tag{2.14}$$

where $\xi(\lambda)$ is an orthogonal increment process and $A_{t,T}(\lambda) \approx F(\frac{t}{T}, \lambda)$, with $A_{t,T}$ and $F(\frac{t}{T}, .)$ being the frequency domain representations of the filters generated by $a_{t,T}$ and $f(\frac{t}{T}, .)$, respectively. These are also called as the time-frequency spectrum of the process. This representation is important in this thesis because there exists a similar decomposition of LSGPs introduced in Section 3.1.1.

As seen in (2.14), a LSP in Dahlhaus' sense is characterized by an approximate filter F which may be seen as a power spectral density of the stationary process that is approximately equal to the kernel centered around some point. In [38, 39], this fact is exploited to find the best basis for the estimation of the locally stationary covariance. An approach for the estimation of the time-frequency content of a special kind of LSPs can be found in [40]. In [41], this approximate filter is used to construct a local Whittle likelihood estimator. Although these works lead to well-established techniques in the domain, their generalization to the graph domain is intricate. This is because these filters are assumed to be centered somewhere in $[0,1]$ and not just on the discrete points $\frac{t}{T}$ in question. However, this is not immediately applicable to graph processes, due to their irregular structures.

Although these two definitions may seem distinct, a common approach in the time series analysis literature is to use data tapering or windowing to get a sample estimate, which can be seen in the works [9, 42–44]. For a given center where the process is approximated by a stationary time series, the usage of data tapers diminishes the effect of the samples that are closer to the boundary of its support. A similar notion is used in this thesis, where the membership functions assess the validity of the given stationary component model. In this work, the membership functions are learnt from data.

2.5 Vertex-Varying Graph Signal Processing

In this chapter we will analyse some related techniques and highlight the differences between this work and the others. By definition, LSGPs exhibit a node-varying nature, hence we will compare the recent advances in the literature with the works related to vertex-varying graph signal processing. First, we will give examples of node-varying graph filters and show how our work relates to them. Then we will look at another deterministic notion of vertex-frequency spectrum and explain why this is different from our construction in Section 3.1.1. Second, we will look at related stochastic process definitions in GSP which are somewhat related to our work.

Contrary to the filters defined in (2.8a), some filters in the GSP literature are more elaborate. These are called node-varying graph filters as shown in the works [45–47]. In these works, there are several definitions of more intricate filters however the most related one to our discussion is the node-varying graph filter whose definition is given in (2.15) [47].

$$\mathbf{y} = \sum_{k=0}^{K-1} \mathbf{G}_k \mathbf{S}^k \mathbf{x} \tag{2.15}$$

Here $\mathbf{S}$ is the structure matrix mentioned in Section 2.2 and $\mathbf{G}_k$ are diagonal matrices encoding the node-varying information of polynomial kernels.

Notably, this filtering operation is closely related to ours for creating a LSGP and suprisingly, in one of the works [48] considering these filters, a non-stationary process is clearly hinted. However, in the rest of the paper signals are assumed to be deterministic. Another point that is different from our work in the definition of these filters in [47, 48] is that there is no mention of smoothness of membership functions. Our result (3.4) in Theorem 1 is proved in [49] for polynomial kernels, however in our case we prove that transitions of these kernels are also smooth if diagonal matrices are smooth on the graph and our assumption is more general as it can potentially incorporate any type of graph kernels.

In classical signal processing, it is common practice to utilize several windows localized on some time value to measure the frequency characteristics around that neighborhood [50, 51] which are used in the analysis of signals with Short Time Fourier

Transform (STFT) [52]. In GSP, this practice is generalized by localizing filters in both vertex and frequency domains [53–55]. In [56], several localized kernels with different scales in the frequency domain are used to generate a vertex-frequency spectrum of a graph signal which is a mere generalization of STFT. This is similar to the concept of vertex-frequency spectrum of LSGPs, but with slight differences. For LSGPs, the vertex-frequency spectrum corresponds to a matrix that multiplicatively interacts with the eigenvectors of the graph Laplacian. This interaction alters the behavior of eigenvectors on each vertex without the need for defining a windowing function. Consequently, an operation akin to the STFT for determining the spectrum might not be available.

As emphasized in Section 2.3, WSS processes are vertex-invariant meaning that non-stationary processes have vertex variance or have more intricate generators. A simple form of this variance is to assume that processes are stationary on their associated subgraphs, contrary to our definition which assumes several stationary kernels on the supergraph. This type of process is introduced in [11] and is called a Piecewise Stationary Process. An important problem in this work is to find the pieces, i.e. subgraphs, and the algorithm given in the same work [11] aims to extract those subgraphs, which uses process values and groups together the vertices with the same activation pattern throughout the realizations of the process. This contrasts with our work on estimating the subgraphs, where we utilize the covariance matrix of the process.

The term "local stationarity" occurs in several occasions in the GSP literature in the works of Girault et al. [54, 57] and Scalzo et al. [19]. In [57], the criterion for being a LSP is rather strict since the local power spectrum in (2.16) of the signal should be the same for every localization of the signal using localized atoms $\mathbf{g}_{i,m}$ where i, m are the indices denoting the vertex number and scale of the localized atoms. The scale in this discussion is usually controlled by shifting frequency bands of a single kernel. As explained in Section 2.2, the spread of the function should decrease as frequency increases because the kernel rapidly changes.

$$\mathbf{S}_{\mathbf{x}}(i,m) = E[| < \mathbf{x}, \mathbf{g}_{i,m} > |^2] \tag{2.16}$$

Thus, even for WSS signals this condition may not be satisfied as pointed out in

[57]. However, this definition of spectrum is useful for estimating the power spectral density of a WSS signal.

In [19], a stochastic process $\mathbf{x}$ is called LSP if, within each specified neighborhood $\mathcal{V}_k$, the mean, variance and correlations between the vertices is constant. This condition is given in (2.17).

$$E[\mathbf{x}_v] = \mathbf{m}_k, \quad E[\mathbf{x}_v^2] = \sigma_k^2, \quad E[\mathbf{x}_v \mathbf{x}_w] = \rho_k, \quad \forall v, w \in \mathcal{V}_k \tag{2.17}$$

Although this condition is weaker than that of Girault, which assumes that (2.16) is a constant mapping, it coincides with our definition in the case where the weight function $\mathbf{W}$ is a constant function with constant membership within $\mathcal{V}_k$. Hence, it is still more strict than our definition.

CHAPTER 3

LOCALLY STATIONARY GRAPH PROCESSES

In this chapter, we will introduce the proposed LSGP model, together with a comprehensive analysis.

3.1 Locally Stationary Graph Processes

We propose a locally stationary graph process model of the form

$$\mathbf{x} = \sum_{k=1}^{K} \mathbf{G}_k \mathbf{U}_{\mathcal{G}} h_k(\mathbf{\Lambda}_{\mathcal{G}}) \mathbf{U}_{\mathcal{G}}^T \mathbf{w} \tag{3.1}$$

where the process $\mathbf{x} \in \mathbb{R}^N$ of interest is obtained by filtering a white noise process $\mathbf{w} \in \mathbb{R}^N$ of unit variance. Here, the overall filtering operation on the white process is expressed in terms of K graph filters of the form $\mathbf{U}_{\mathcal{G}} h_k(\mathbf{\Lambda}_{\mathcal{G}}) \mathbf{U}_{\mathcal{G}}^T$, each of which is defined by the kernel $h_k(\lambda)$. For each k, the term $\mathbf{U}_{\mathcal{G}} h_k(\mathbf{\Lambda}_{\mathcal{G}}) \mathbf{U}_{\mathcal{G}}^T \mathbf{w}$ is an individual WSS graph process, which forms a "component" of our locally stationary graph process model. The matrices $\mathbf{G}_k \in \mathbb{R}^{N \times N}$ are diagonal matrices representing the "membership" of the overall process $\mathbf{x}$ with respect to each component WSS process: For each n-th node, the n-th diagonal entry $\mathbf{G}_k(n, n)$ indicates how much the k-th component process contributes to the overall process $\mathbf{x}$. Note that an analogy can be drawn between (3.1) and the classical locally stationary time process model in (2.13), such that the filter coefficients $a_{t,T}$ in (2.13) would correspond to the graph filters $\mathbf{U}_{\mathcal{G}} h_k(\mathbf{\Lambda}_{\mathcal{G}}) \mathbf{U}_{\mathcal{G}}^T$ in (3.1). While the local behavior of the process model is captured by the dependence of the filters $a_{t,T}$ on the time instant t in (2.13), in (3.1) it is captured by the membership matrices $\mathbf{G}_k$ whose n-th entries define the local behavior of the process at the n-th graph node.

In (3.1) we do not make any general assumptions about the spectral characteristics of the graph filters $h_k(\cdot)$ defining the model. In return, in order to provide the model with a meaningful notion of local stationarity, one must ensure that the statistics of the process change smoothly between neighboring nodes. We achieve this by imposing that the membership functions of the process vary slowly on the graph: Defining the vectorized form $\mathbf{g}_k \in \mathbb{R}^N$ of the diagonal matrices $\mathbf{G}_k$ such that $\mathbf{g}_k(n) = \mathbf{G}_k(n,n)$, the vector $\mathbf{g}_k$ can be regarded as a membership function on the graph, which identifies how much each graph node conforms to the k-th component process model. An upper bound on the term $\mathbf{g}_k^T \mathbf{L}_\mathcal{G} \mathbf{g}_k$ then determines the rate at which the membership function $\mathbf{g}_k$ varies on the graph. This motivates the following definition of locally stationary graph processes (LSGP):

Definition 2 (Locally Stationary Graph Process)**.** *A stochastic graph signal* $\mathbf{x}$ *is called a locally stationary graph process (LSGP) with variation rate* C*, if it can be expressed as*

$$\mathbf{x} = \sum_{k=1}^{K} \mathbf{G}_k \mathbf{U}_\mathcal{G} h_k(\mathbf{\Lambda}_\mathcal{G}) \mathbf{U}_\mathcal{G}^T \mathbf{w} \tag{3.2}$$

such that $\mathbf{g}_k^T \mathbf{L}_\mathcal{G} \mathbf{g}_k \leq C$ *and* $\|h_k(\mathbf{\Lambda}_\mathcal{G})\|_F^2 = 1$*, for* $k = 1, \ldots, N$.

Let $\mathbf{h}_k \in \mathbb{R}^N$ denote the vectorized form of the diagonal entries of $h_k(\mathbf{\Lambda}_\mathcal{G})$ such that $\mathbf{h}_k(n) = h_k(\mathbf{\Lambda}_\mathcal{G})(n,n)$. In the above definition, the normalization condition $\|h_k(\mathbf{\Lambda}_\mathcal{G})\|_F^2 = 1$ (or equivalently $\|\mathbf{h}_k\| = 1$) on the filter kernels is imposed in order to avoid the scale ambiguity arising from the product of the membership values $\mathbf{g}_k$ with the amplitudes of the kernels $\mathbf{h}_k$.

In this construction, given an unweighted finite ring graph with $T + Q - 1$ nodes, it is possible to obtain a special instance of the time process model in (2.13) as the last T observations on the graph, with indices ranging from 0 to $T - 1$. Therefore, for each time instant, $a_{t,T}$ is restricted to have Q coefficients and be a causal FIR filter. Under these conditions, we impose that $a_{t,T}$ and f satisfy the weak assumptions presented in [9]. To recover this process using (3.2), let the membership functions and component kernels be defined as per (3.3). It is assumed that membership functions are 0 outside the nodes corresponding to the $[0, T-1]$ interval.

$$
\begin{aligned}
&\tilde{h}_k(\lambda) = \lambda^{k-1} \\
&h_k(\lambda) = \tilde{h}_k(\lambda) / \| \tilde{\mathbf{h}}_k \left(\Lambda_{\mathcal{G}} \right) \|_2 \\
&\mathbf{g}_k(t) = \| \tilde{\mathbf{h}}_k \left(\Lambda_{\mathcal{G}} \right) \|_2 \cdot a_{t,T}(k-1) \\
&k \in \{1, 2, \ldots, Q\} \text{ and } t \in \{0, 1, \ldots, T-1\}
\end{aligned}
\tag{3.3}
$$

The variation rate of this process is bounded, if the number of samples, T, is finite. This is due to the fact that the kernel, f, has bounded variation over the $(0, 1]$ interval and the deviation of the FIR filter coefficients $a_{t,T}(j)$ from f is assumed to be bounded at each node instant. Therefore, the LSGP defined on a ring graph according to (3.3) is equivalent to the LSP in (2.13), when $a_{t,T}(i)$ are chosen as the coefficients of a causal FIR filter with Q-taps. This shows that the proposed LSGP model (3.2) generalizes locally stationary time processes to graph domains.

3.1.1 Vertex-Frequency Spectrum

In classical time series analysis, a locally stationary time process is associated with a time-frequency spectrum $A_{t,T}$ as in (2.14). We now show that the notion of vertex-frequency spectrum can be generalized to LSGPs as well.

Theorem 1 (Vertex-Frequency Spectrum)**.** *A locally stationary graph process with variation rate C can be expressed as $\mathbf{x} = \mathbf{H}\mathbf{w}$, where the filter*

$$
\mathbf{H} = \sum_{k=1}^{K} \mathbf{G}_k \mathbf{U}_{\mathcal{G}} h_k(\mathbf{\Lambda}_{\mathcal{G}}) \mathbf{U}_{\mathcal{G}}^T
$$

can be written in the form

$$
\mathbf{H} = (\mathbf{U}_{\mathcal{G}} \circ \mathbf{M}) \mathbf{U}_{\mathcal{G}}^T \tag{3.4}
$$

where $\mathbf{M} = \sum_{k=1}^{K} \mathbf{g}_k \mathbf{h}_k^T$. Here the matrix $\mathbf{M} \in \mathbb{R}^{N \times N}$ identifies the vertex-frequency spectrum of the process, such that $\mathbf{M}(n, i)$ gives the local spectrum for the i-th graph frequency $\lambda_{\mathcal{G}}(i)$ at graph node n. For each graph frequency $\lambda_{\mathcal{G}}(i)$, the local spectrum $\mathbf{M}(\cdot, i)$ has bounded variation over the whole graph when regarded as a graph signal, such that the total variation of the local spectra is upper bounded as

$$
tr(\mathbf{M}^T \mathbf{L}_{\mathcal{G}} \mathbf{M}) \leq K^2 C.
$$

Proof. We first present the following lemma, which is due to [58].

Lemma 1. *For any* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N\times M}$ *and diagonal matrices* $\mathbf{D}, \mathbf{E}$*, the following equality holds*

$$\mathbf{D}(\mathbf{A} \circ \mathbf{B})\mathbf{E} = \mathbf{A} \circ (\mathbf{DBE}).$$

Taking $\mathbf{D} = \mathbf{G}_k$, $\mathbf{A} = \mathbf{U}_\mathcal{G}$, $\mathbf{B} = \mathbf{1}_{N\times N}$ and $\mathbf{E} = h_k(\mathbf{\Lambda}_\mathcal{G})$ in Lemma 1, we have

$$\begin{aligned}\mathbf{H} &= \sum_{k=1}^{K} \mathbf{G}_k \mathbf{U}_\mathcal{G} h_k(\mathbf{\Lambda}_\mathcal{G}) \mathbf{U}_\mathcal{G}^T = \sum_{k=1}^{K} \mathbf{G}_k (\mathbf{U}_\mathcal{G} \circ \mathbf{1}_{N\times N}) h_k(\mathbf{\Lambda}_\mathcal{G}) \mathbf{U}_\mathcal{G}^T \\ &= \sum_{k=1}^{K} \left(\mathbf{U}_\mathcal{G} \circ (\mathbf{G}_k\ \mathbf{1}_{N\times N}\ h_k(\mathbf{\Lambda}_\mathcal{G}))\right) \mathbf{U}_\mathcal{G}^T \\ &= \left(\mathbf{U}_\mathcal{G} \circ \left(\sum_{k=1}^{K} \mathbf{G}_k\ \mathbf{1}_{N\times N}\ h_k(\mathbf{\Lambda}_\mathcal{G})\right)\right) \mathbf{U}_\mathcal{G}^T\end{aligned} \tag{3.5}$$

where the third and the fourth equalities follow respectively from Lemma 1 and the linearity of the Hadamard product. Defining

$$\mathbf{M} = \sum_{k=1}^{K} \mathbf{G}_k\ \mathbf{1}_{N\times N}\ h_k(\mathbf{\Lambda}_\mathcal{G}) \tag{3.6}$$

we arrive at the equality in (3.4).

The matrix $\mathbf{M}$ in (3.6) can equivalently be written as $\mathbf{M} = \sum_{k=1}^{K} \mathbf{g}_k \mathbf{h}_k^T$. We then observe that each i-th row of $\mathbf{M}$ is given by $\sum_k \mathbf{g}_k(i)\mathbf{h}_k^T$, hence represents the overall spectrum at node i resulting from the contributions of all individual spectra $\mathbf{h}_k$ weighted by the membership $\mathbf{g}_k(i)$ of the i-th node to the k-th model. Therefore, the matrix $\mathbf{M}$ provides the vertex-frequency spectrum of the locally stationary graph process $\mathbf{x}$.

We next bound the variation of the local spectra on the graph as follows. Writing

$$\mathrm{tr}(\mathbf{M}^T \mathbf{L}_\mathcal{G} \mathbf{M}) = \sum_{i,j=1}^{K} \mathrm{tr}(\mathbf{h}_i \mathbf{g}_i^T \mathbf{L}_\mathcal{G} \mathbf{g}_j \mathbf{h}_j{}^T) = \sum_{i,j=1}^{K} \mathbf{g}_i^T \mathbf{L}_\mathcal{G} \mathbf{g}_j \mathrm{tr}(\mathbf{h}_i \mathbf{h}_j^T) \tag{3.7}$$

and noting that $\text{tr}(\mathbf{h}_i\mathbf{h}_j^T) = \langle \mathbf{h}_i, \mathbf{h}_j \rangle \triangleq \mathbf{h}_i^T\mathbf{h}_j$, we have

$$\begin{aligned} \text{tr}(\mathbf{M}^T\mathbf{L}_{\mathcal{G}}\mathbf{M}) = |\text{tr}(\mathbf{M}^T\mathbf{L}_{\mathcal{G}}\mathbf{M})| &= \left| \sum_{i,j=1}^{K} \mathbf{g}_i^T\mathbf{L}_{\mathcal{G}}\mathbf{g}_j \, \langle \mathbf{h}_i, \mathbf{h}_j \rangle \right| \\ \leq \sum_{i,j=1}^{K} \left| \mathbf{g}_i^T\mathbf{L}_{\mathcal{G}}\mathbf{g}_j \right| |\langle \mathbf{h}_i, \mathbf{h}_j \rangle| &\leq C \sum_{i,j=1}^{K} |\langle \mathbf{h}_i, \mathbf{h}_j \rangle| \leq K^2 C \end{aligned} \tag{3.8}$$

where the last two inequalities follow from the Cauchy-Schwarz inequality and the fact that $\mathbf{L}_{\mathcal{G}}$ is a positive semi-definite matrix.

□

An immediate implication of Theorem 1 is that the vertex-frequency spectrum $\mathbf{M}$ of a locally stationary graph process is the counterpart of the time-frequency spectrum $A_{t,T}$ for locally stationary time processes; hence, $\mathbf{M}$ defines a vertex-varying spectrum for LSGPs. Noticing that a WSS graph process is always an LSGP due to Definition 1, it is easy to verify that the vertex-frequency spectrum $\mathbf{M}$ of a WSS graph process takes the trivial form of a rank-1 matrix with identical row vectors.

An illustration of the ideas presented here is shown in Fig. 3.1. First a LSGP has been constructed on a randomly generated graph (see Section 6.1.1 for details of graph construction.). The component kernels, which are cubic polynomial filters with low-pass characteristics in this case, are shown in Fig. 3.1(b). We let membership functions be constant on a subgraph and decrease smoothly towards other subgraphs as shown in Fig. 3.1(a). Since the area where the membership functions blend is relatively small with respect to the subgraphs the vertex-frequency behaviour of separate polynomials is observable in the vertex-frequency spectrum as seen in Fig. 3.1(c). Some example realizations generated with the given model can be seen Fig. 3.1(d)-3.1(e). Notice the variation of the graph signal on the nodes, demonstrating the variation of the membership functions.

3.1.2 Extension and Restriction of LSGPs

In the analysis of graph signals, a question of interest is whether a signal model defined on a graph can be extended to larger graphs in a mathematically consistent way, e.g. as would be required in applications where new nodes are added to the graph.

Similarly, one may be interested in restricting the signal model to a subgraph of the original graph as well, e.g. due to node removal. In this section, we discuss the extension and restriction of locally stationary graph process models, and show that the family of LSGPs are sufficiently comprehensive so as to permit their extension and restriction to larger and smaller graphs by still remaining in the class of LSGP models.

Let us consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ with a given subgraph $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s, \mathbf{W}_s)$. The relation between the subgraph $\mathcal{G}_s$ and the supergraph $\mathcal{G}$ can be represented through an inclusion map $\iota = (\iota_v, \iota_e, \iota_w) : \mathcal{G}_s \hookrightarrow \mathcal{G}$, where ι_v, ι_e, and ι_w denote the inclusion maps defined over the vertices, edges, and edge weights, respectively. One can then define a binary selection matrix $\mathbf{S}_s \in \{0,1\}^{|\mathcal{V}_s| \times |\mathcal{V}|}$ such that $\mathbf{S}_s(v_s, v) = 1$ whenever $\iota_v(v_s) = v, \forall v_s \in \mathcal{V}_s$ for a given enumeration of the vertices in $\mathcal{V}_s$ and $\mathcal{V}$.

3.1.2.1 Extension of LSGPs

We consider an LSGP $\mathbf{x}_s$ on the subgraph $\mathcal{G}_s$ defined by the kernels $\mathbf{h}_{s,k}$ and the membership functions $\mathbf{g}_{s,k}$ for $k = 1, \ldots, K$. Due to Theorem 1, the process $\mathbf{x}_s$ can be expressed in terms of its vertex-frequency spectrum as

$$\mathbf{x}_s = \left(\mathbf{U}_{\mathcal{G}_s} \circ \sum_{k=1}^{K} \mathbf{g}_{s,k} \mathbf{h}_{s,k}^T \right) \mathbf{U}_{\mathcal{G}_s}^T \, \mathbf{w}_s \tag{3.9}$$

where $\mathbf{w}_s \in \mathbb{R}^{|\mathcal{V}_s|}$ is a unit-variance white process. In order to extend the process $\mathbf{x}_s$ to the supergraph $\mathcal{G}$, in addition to the inclusion map ι in the vertex domain, we will also make use of an inclusion map in the frequency domain, defined as $\hat{\iota} : \{1, 2, \ldots, |\mathcal{V}_s|\} \hookrightarrow \{1, 2, \ldots, |\mathcal{V}|\}$. The inclusion map $\hat{\iota}$ determines how the frequencies in the spectrum of $\mathbf{L}_{\mathcal{G}_s}$ should relate to those of $\mathbf{L}_{\mathcal{G}}$. We maintain a generic setting by treating $\hat{\iota}$ as an arbitrary injection, whose selection is in fact a matter of choice among several possible strategies. The spectral selection matrix $\hat{\mathbf{S}} \in \{0,1\}^{|\mathcal{V}_s| \times |\mathcal{V}|}$ associated with $\hat{\iota}$ is given by $\hat{\mathbf{S}}(v_s, v) = 1$ for $\hat{\iota}(v_s) = v$.

We can then define the extension of the process model $\mathbf{x}_s$ to the supergraph $\mathcal{G}$ as

$$\mathbf{x} = \left(\mathbf{U}_{\mathcal{G}} \circ \sum_{k=1}^{K} \mathbf{S}_s^T \mathbf{g}_{s,k} \mathbf{h}_{s,k}^T \hat{\mathbf{S}} \right) \mathbf{U}_{\mathcal{G}}^T \mathbf{w}. \tag{3.10}$$

The extended process model is thus obtained by extending the membership functions

and the kernels to the graph $\mathcal{G}$, respectively as $\mathbf{g}_k = \mathbf{S}_s^T \mathbf{g}_{s,k}$ and $\mathbf{h}_k = \hat{\mathbf{S}}^T \mathbf{h}_{s,k}$. Noticing that the choice of $\hat{\iota}$, and thus $\hat{\mathbf{S}}$, determines the vertex-frequency spectrum of the extended process $\mathbf{x}$, the extension operation in (3.10) is observed to define an injective map between the set of LSGPs on the subgraph $\mathcal{G}_s$ and the set of LSGPs on $\mathcal{G}$ by retaining the model order K. The adoption of the normalized graph Laplacian in our process model provides a convenient basis for the extension procedure in (3.10) due to the boundedness of its eigenvalues. It can be verified that the extended process x takes the value 0 on the nodes in $\mathcal{V} \setminus \mathcal{V}_s$.

The extension from multiple subgraphs without data loss is also possible if the subgraphs are disjoint. Let $\{\mathcal{G}_{s_i}\}_{i=1}^S$ be the subgraphs of $\mathcal{G}$ such that $\mathcal{V}_{s_i} \cap \mathcal{V}_{s_j} = \emptyset$, $i \neq j$ and let us define LSGPs $\mathbf{x}_{s_i}$ corresponding to the ith subgraph. Assume that each x_{s_i} is generated by the membership function and component kernel pairs $\left(\{\mathbf{g}_{s_i,k}\}_{k=1}^K, \{\mathbf{h}_{s_i,k}\}_{k=1}^K\right)$. Moreover, for each i, we have a vertex selection matrix $\mathbf{S}_i$ corresponding to $\mathcal{G}_{s_i}$ and a spectral selection matrix $\hat{\mathbf{S}}_i$ associated by the inclusion $\hat{\iota}_i$. Under these constraints and definitions, the joint extension of the processes on the subgraphs can be expressed as

$$\mathbf{x} = \left(\mathbf{U}_{\mathcal{G}} \circ \sum_{i=1}^{S} \sum_{k=1}^{K} \mathbf{S}_{s_i,k}^T \mathbf{g}_{s_i,k} \mathbf{h}_{s_i,k}^T \hat{\mathbf{S}}_{s_i,k} \right) \mathbf{U}_{\mathcal{G}}^T \mathbf{w}. \tag{3.11}$$

The constraint that the subgraphs are disjoint allows for the extension without data loss by virtue of each subgraph having a mutually exclusive selection matrix. Consequently, the rows of the vertex-frequency spectrum are filled without overlap. Similar to the extension from a single subgraph, in this case, we can obtain membership functions and component kernels on the supergraph $\mathcal{G}$ as $\mathbf{g}_{i(K-1)+k+1} = \mathbf{S}_{s_i}^T \mathbf{g}_{s_i,k}$ and $\mathbf{h}_{i(K-1)+k+1} = \hat{\mathbf{S}}_{s_i}^T \mathbf{h}_{s_i,k}$. This approach generates at most KS membership functions on the supergraph.

If the selected subgraphs have overlapping nodes, i.e., $\exists i, j,\ \mathcal{V}_{s_i} \cap \mathcal{V}_{s_j} \neq \emptyset$, the extension in (3.11) introduces an LSGP on $\mathcal{G}$ defined by the extended membership functions and component kernels on the supergraph. However, this may lead to data loss, if the frequency content of the subgraph processes is supported across all graph frequencies. In this case, the composition of restriction and extension will not result in

preservation of the original processes $\mathbf{x}_{s_j}$ and $\mathbf{x}_{s_i}$ as highlighted in Remark 1.

3.1.2.2 Restriction of LSGPs

We next consider an LSGP $\mathbf{x}$ on the supergraph $\mathcal{G}$ given by

$$\mathbf{x} = \left(\mathbf{U}_{\mathcal{G}} \circ \sum_{k=1}^{K} \mathbf{g}_k \mathbf{h}_k^T \right) \mathbf{U}_{\mathcal{G}}^T \mathbf{w}. \tag{3.12}$$

Based on the node and vertex mappings ι and $\hat{\iota}$, we define the restriction of $\mathbf{x}$ to the subgraph $\mathcal{G}_s$ as

$$\mathbf{x}_s = \left(\mathbf{U}_{\mathcal{G}_s} \circ \sum_{k=1}^{K} \mathbf{S}_s \mathbf{g}_k \mathbf{h}_k^T \hat{\mathbf{S}}^T \right) \mathbf{U}_{\mathcal{G}_s}^T \mathbf{w}_s. \tag{3.13}$$

Similarly to the extension procedure, we define the model parameters of the restricted process $\mathbf{x}_s$ as $\mathbf{g}_{s,k} = \mathbf{S}_s \mathbf{g}_k$ and $\mathbf{h}_{s,k} = \hat{\mathbf{S}} \mathbf{h}_k$, which identify its vertex-frequency spectrum. Hence, the restriction operation defines a surjective map between the set of LSGPs on the supergraph $\mathcal{G}$ and the set of LSGPs on the subgraph $\mathcal{G}_s$. The formulation in (3.13) reveals that the model order of the restricted process is at most K; however, due to the potential loss of information in restriction, the resulting process may occasionally be represented with a smaller order.

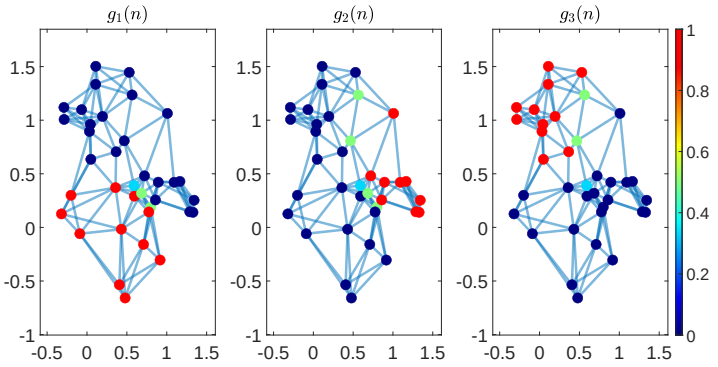
It should be noted that, restricting a process to a subgraph of $\mathcal{G}$ can potentially lead to data loss in the process residing on the graph. This data loss may be prevented by restricting the process on several subgraphs possibly having overlapping nodes.

For most graph topologies, the extension procedure defined in (3.10) and the restriction procedure in (3.13) do not preserve the signals located on the vertices of the subgraph. A notable exception to this occurs when the graph is composed of several connected components. Under this condition, the eigenvector matrix of the supergraph is block diagonal, where each block corresponds to the eigenvector matrix of a connected component within the graph. Therefore, the restriction and extension procedures will preserve the signals on the connected components, as long as the frequency selection matrix retains the frequencies of the connected component properly.

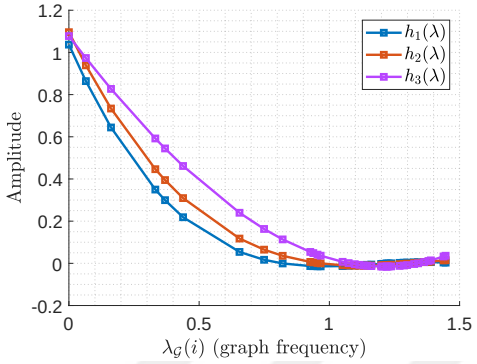
Inspired by this discussion, the accuracy of the extension and restriction process can be associated with the eigenvalue gap of the graph Laplacian matrix of the discon-

nected graph, particularly when considering the problem on subgraphs that are connected with edges with relatively low weights. Thus, in certain applications, it becomes possible to learn the process on the subgraph and extend it to a supergraph by the simple procedure in (3.10), with some accuracy.

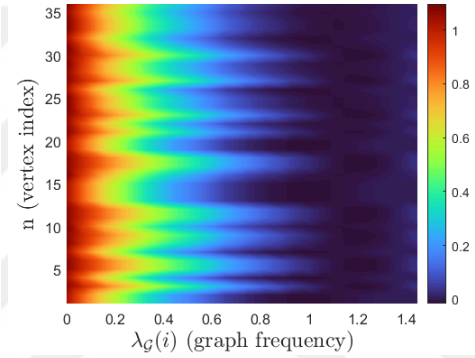
Remark 1. *Following the definitions of the extension and restriction operations, a pertinent question is whether the extension of a process $\mathbf{x}_s$ from $\mathcal{G}_s$ to $\mathcal{G}$, followed by its restriction back to $\mathcal{G}_s$ preserves the original process $\mathbf{x}_s$. Under the mild assumption that the matrix $(\mathbf{S}_s\mathbf{U}_{\mathcal{G}}\hat{\mathbf{S}}^T) \circ \mathbf{U}_{\mathcal{G}_s}$ has no zero entries, it is easy to show that the consequent application of the operations in* (3.10) *and* (3.13) *lead to an identity morphism on the set of LSGPs on $\mathcal{G}_s$ with model order K, hence ensures that the process $\mathbf{x}_s$ is preserved. The assumption that $(\mathbf{S}_s\mathbf{U}_{\mathcal{G}}\hat{\mathbf{S}}^T) \circ \mathbf{U}_{\mathcal{G}_s}$ should have nonzero entries is very likely to be met for irregular graph topologies consisting of a single connected component, since the eigenvectors of the graph Laplacian do not typically contain zero entries in this case.*



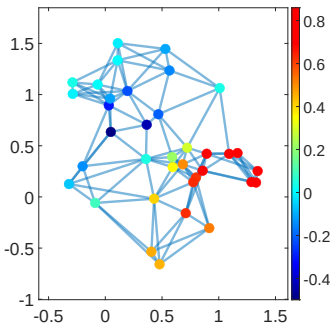
(a) Membership functions



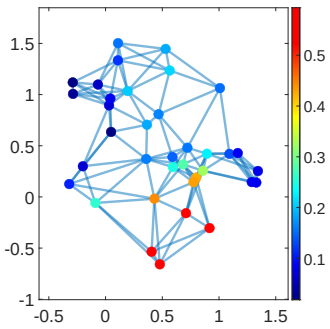
(b) Corresponding components



(c) Vertex-frequency spectrum



(d) A realization of the generated process



(e) Another realization of the generated process

Figure 3.1: LSGP Model

CHAPTER 4

LEARNING LSGP MODELS

In Section 3.1, we proposed a model for locally stationary graph processes and examined some of its properties. We next study the problem of learning LSGPs in this chapter. We propose a framework for computing an LSGP model from a set $\{\mathbf{x}^l\}_{l=1}^L$ of L graph signals that are considered as realizations of a process $\mathbf{x}$. Our aim is then to compute a process model of the form (3.2). In many practical applications, the graph signals of interest can be only partially observed; hence, we address a flexible setting where the given realizations $\mathbf{x}^l \in \mathbb{R}^N$ of the process may contain missing values.

4.1 Polynomial Approximation of WSS Components

In this section, we will propose an optimization problem to obtain an LSGP model whose component processes are generated from the polynomial kernels.

4.1.1 Algorithm

We consider that the value $\mathbf{x}^l(i)$ of the realization is known at a subset of the graph nodes $i \in \{1, \ldots, N\}$. Let us denote as I^l the index set of the nodes i where $\mathbf{x}^l(i)$ is known. In order to compute an LSGP model, we first obtain a sample covariance estimate $\hat{\mathbf{C}}_\mathbf{x}$ of the covariance matrix $\mathbf{C}_\mathbf{x}$ of the process from the available observations. Our approach is then based on learning a process model by fitting the parameters of the LSGP in (3.2) to the covariance estimate $\hat{\mathbf{C}}_\mathbf{x}$. The covariance matrix of the graph process $\mathbf{x} = \mathbf{H}\mathbf{w}$ as defined in Theorem 1 is given by

$$\mathbf{C}_\mathbf{x} = E[\mathbf{x}\mathbf{x}^T] = \mathbf{H}\mathbf{H}^T \tag{4.1}$$

assuming that $\mathbf{w}$ is a zero-mean white process with unit variance. We then propose to learn the process model by solving the following optimization problem:

$$\begin{aligned} &\underset{\{\mathbf{g}_k\}_{k=1}^K, \{\mathbf{h}_k\}_{k=1}^K}{\text{minimize}} \; ||\hat{\mathbf{C}}_{\mathbf{x}} - \mathbf{H}\mathbf{H}^T||_F^2 + \mu_1 \operatorname{tr}(\mathbf{G}^T \mathbf{L}_{\mathcal{G}} \mathbf{G}) \\ &\text{subject to } \mathbf{G} = [\mathbf{g}_1 \; \ldots \; \mathbf{g}_K], \quad \mathbf{H} = \sum_{k=1}^{K} \mathbf{G}_k \mathbf{U}_{\mathcal{G}} h_k(\mathbf{\Lambda}_{\mathcal{G}}) \mathbf{U}_{\mathcal{G}}^T. \end{aligned} \tag{4.2}$$

The above optimization problem is motivated by the local stationarity definition in Definition 2. The first term in the objective function enforces the covariance matrix $\mathbf{C}_{\mathbf{x}} = \mathbf{H}\mathbf{H}^T$ of the learnt process model to fit the initial empirical estimate $\hat{\mathbf{C}}_{\mathbf{x}}$. Gathering all membership functions $\mathbf{g}_k$ in the matrix $\mathbf{G} \in \mathbb{R}^{N \times K}$, the second term aims to reduce the variation rate C of the locally stationary process as much as possible, so that the process statistics change smoothly throughout the graph.

One can observe that the optimization problem in (4.2) is difficult to solve as it is nonconvex and it involves $2NK$ optimization variables. In order to put the problem in a more tractable form, we first constrain the graph kernels $h_k(\cdot)$ to be polynomial functions, which is a common choice due to its various convenient properties such as good vertex-domain localization [47]. The entries of the spectrum vector $\mathbf{h}_k$ are then of the form

$$\mathbf{h}_k(i) = \sum_{q=0}^{Q-1} b_{q,k} \lambda_{\mathcal{G}}^q(i) \tag{4.3}$$

where $\lambda_{\mathcal{G}}^q(i)$ denotes the q-th power of the i-th graph frequency $\lambda_{\mathcal{G}}(i)$; and $b_{q,k}$ are polynomial coefficients. Let us define the polynomial coefficient vector

$$\mathbf{b}_k = [b_{0,k} \; b_{1,k} \; \ldots \; b_{Q-1,k}]^T \in \mathbb{R}^Q$$

of the k-th kernel, as well as the overall coefficient vector $\mathbf{b} = [\mathbf{b}_1^T \; \mathbf{b}_2^T \; \ldots \; \mathbf{b}_K^T]^T \in \mathbb{R}^{QK}$. Let also $\mathbf{g} = [\mathbf{g}_1^T \; \mathbf{g}_2^T \; \ldots \; \mathbf{g}_K^T]^T \in \mathbb{R}^{NK}$ denote the vectorized form of the matrix $\mathbf{G}$. Then we propose to relax the nonconvex problem (4.2) into a convex one, by introducing the new optimization variables

$$\mathbf{\Gamma} = \mathbf{g}\mathbf{g}^T \in \mathbb{R}^{NK \times NK}, \qquad \mathbf{B} = \mathbf{b}\mathbf{b}^T \in \mathbb{R}^{QK \times QK}. \tag{4.4}$$

In Section 4.1.2, we show that the term $\mathbf{H}\mathbf{H}^T$ can be directly expressed as a function of $\mathbf{\Gamma}$ and $\mathbf{B}$; and the term $\operatorname{tr}(\mathbf{G}^T \mathbf{L}_{\mathcal{G}} \mathbf{G})$ can be written as a linear function of $\mathbf{\Gamma}$. We

can thus define the functions $f_1(\mathbf{\Gamma}, \mathbf{B}) = ||\hat{\mathbf{C}}_{\mathbf{x}} - \mathbf{H}\mathbf{H}^T||_F^2$ and $f_2(\mathbf{\Gamma}) = \text{tr}(\mathbf{G}^T\mathbf{L}_{\mathcal{G}}\mathbf{G})$ representing the first and the second terms of the objective in (4.2). Meanwhile, for the decompositions in (4.4) to be valid, the matrices $\mathbf{\Gamma}$ and $\mathbf{B}$ need to be rank-1 and positive semi-definite. We thus propose to relax the original problem (4.2) into the optimization problem

$$\begin{aligned} \underset{\Gamma, B}{\text{minimize}}\ & f_1(\mathbf{\Gamma}, \mathbf{B}) + \mu_1 f_2(\mathbf{\Gamma}) + \mu_2 \text{tr}(\mathbf{B}) + \mu_3 \text{tr}(\mathbf{\Gamma}) \\ & \text{subject to } \mathbf{\Gamma} \in \mathbb{S}_+^{NK},\ \ \mathbf{B} \in \mathbb{S}_+^{QK} \end{aligned} \tag{4.5}$$

where $\mathbb{S}_+^n$ denotes the cone of $n \times n$ positive semi-definite matrices and μ_1, μ_2, μ_3 are positive weight parameters. The terms $\text{tr}(\mathbf{B})$ and $\text{tr}(\mathbf{\Gamma})$ in the objective function give the nuclear norms of the positive semi-definite matrices $\mathbf{B}$ and $\mathbf{\Gamma}$, aiming to minimize their ranks by providing a relaxation of the rank-1 constraint.

The term $f_1(\mathbf{\Gamma}, \mathbf{B})$ in (4.5) is quadratic individually in $\mathbf{\Gamma}$ and in $\mathbf{B}$, and the term $f_2(\mathbf{\Gamma})$ is linear in $\mathbf{\Gamma}$. While the overall objective function is not jointly convex in $\mathbf{\Gamma}$ and $\mathbf{B}$, it is convex in only $\mathbf{\Gamma}$ and only $\mathbf{B}$. We propose to minimize the objective iteratively with an alternating optimization procedure. In each iteration, we first fix $\mathbf{\Gamma}$ to optimize $\mathbf{B}$, and then fix $\mathbf{B}$ to optimize $\mathbf{\Gamma}$ with semi-definite programming (SDP) [59] by linearizing the quadratic functions. Once $\mathbf{\Gamma}$ and $\mathbf{B}$ are found in this way, the model parameter vectors $\mathbf{g}$ and $\mathbf{b}$ are computed through their rank-1 approximations via SVD. The filter $\mathbf{H}$ can then be computed from $\mathbf{g}$ and $\mathbf{b}$ using the relations in (3.4) and (4.3). The proposed method for learning LSGP models is summarized in Algorithm 1.

4.1.2 Optimization Problem in Explicit Form

Here we derive the explicit expressions for the terms $f_1(\mathbf{\Gamma}, \mathbf{B})$ and $f_2(\mathbf{\Gamma})$ appearing in our problem formulation in Section 4.1.1. Defining $\mathbf{H}_k = \sum_{q=0}^{Q-1} b_{q,k}\mathbf{L}_{\mathcal{G}}^q$, we have $\mathbf{H} = \sum_{k=1}^{K} \mathbf{G}_k\mathbf{H}_k$, which gives

$$\begin{aligned} \mathbf{H}\mathbf{H}^T &= \sum_{k=1}^{K}\sum_{l=1}^{K} \mathbf{G}_k\mathbf{H}_k\mathbf{H}_l^T\mathbf{G}_l^T \\ &= \sum_{k=1}^{K}\sum_{l=1}^{K}\sum_{q=0}^{Q-1}\sum_{r=0}^{Q-1} \mathbf{G}_k b_{q,k} b_{r,l} \mathbf{L}_{\mathcal{G}}^{q+r}\mathbf{G}_l. \end{aligned} \tag{4.6}$$

Algorithm 1 Learning LSGP models

Input: Graph $\mathcal{G}$, process realizations $\{\mathbf{x}^l\}_{l=1}^L$, number of process components K

Output: LSGP model parameters $\mathbf{g}_k$, $\mathbf{h}_k$, $\mathbf{M}$, $\mathbf{H}$

procedure `learnLSGP`$(\mathcal{G}, \{\mathbf{x}^l\}_{l=1}^L, K)$

1: Compute sample covariance estimate $\hat{\mathbf{C}}_{\mathbf{x}}$ from $\{\mathbf{x}^l\}_{l=1}^L$

2: Estimate second order local filter parameters $\mathbf{\Gamma}, \mathbf{B}$ by solving (4.5) in an alternating manner

3: Find $\mathbf{g}$ and $\mathbf{b}$ with rank-1 decompositions of $\mathbf{\Gamma}$ and $\mathbf{B}$

4: Compute kernels $\{\mathbf{h}_k\}_{k=1}^K$ using (4.3)

5: From $\{\mathbf{g}_k\}_{k=1}^K$ and $\{\mathbf{h}_k\}_{k=1}^K$, compute $\mathbf{M} = \sum_{k=1}^K \mathbf{g}_k \mathbf{h}_k^T$

6: Estimate $\mathbf{H}$ from (3.4)

We proceed by defining $\mathbf{Z}_k = [\mathbf{0}_{N\times N} \ \dots \ \mathbf{I}_{N\times N} \ \dots \ \mathbf{0}_{N\times N}] \in \mathbb{R}^{N\times NK}$ which contains the identity matrix in its k-th block. We then have $\mathbf{Z}_k \mathbf{\Gamma} \mathbf{Z}_l^T = \mathbf{g}_k \mathbf{g}_l^T$. Again using Lemma 1, we set $\mathbf{D} = \mathbf{G}_k$, $\mathbf{E} = \mathbf{G}_l$, $\mathbf{A} = b_{q,k} b_{r,l} \mathbf{L}_{\mathcal{G}}^{q+r}$, $\mathbf{B} = \mathbf{1}_{N\times N}$, and manipulate the resulting equation as in Section 3.1.1 to obtain

$$\begin{aligned} \mathbf{H}\mathbf{H}^T &= \sum_{k=1}^{K}\sum_{l=1}^{K}\sum_{q=0}^{Q-1}\sum_{r=0}^{Q-1} \mathbf{G}_k b_{q,k} b_{r,l} \mathbf{L}_{\mathcal{G}}^{q+r} \mathbf{G}_l \\ &= \sum_{k=1}^{K}\sum_{l=1}^{K}\sum_{q=0}^{Q-1}\sum_{r=0}^{Q-1} (\mathbf{Z}_k \mathbf{\Gamma} \mathbf{Z}_l^T) \circ (b_{q,k} b_{r,l} \mathbf{L}_{\mathcal{G}}^{q+r}). \end{aligned} \tag{4.7}$$

Hence, $\mathbf{H}\mathbf{H}^T$ is shown to be a function of $\mathbf{\Gamma}$. In order to obtain the dependence of $\mathbf{H}\mathbf{H}^T$ on $\mathbf{B}$, we define the matrix

$$\mathbf{Y}_k = [\mathbf{0}_{Q\times Q} \ \dots \ \mathbf{I}_{Q\times Q} \ \dots \ \mathbf{0}_{Q\times Q}]^T \in \mathbb{R}^{QK\times Q}$$

which contains the identity matrix in its k-th block, and the vector $\mathbf{e}_q = [0 \ \dots \ 1 \ \dots 0]^T \in \mathbb{R}^{Q\times 1}$, which contains the value 1 in its q-th entry. We thus obtain the function

$$\begin{aligned} f_1(\mathbf{\Gamma}, \mathbf{B}) = ||\hat{\mathbf{C}}_{\mathbf{x}} - \sum_{k=1}^{K}\sum_{l=1}^{K}\sum_{q=0}^{Q-1}\sum_{r=0}^{Q-1} (\mathbf{Z}_k \mathbf{\Gamma} \mathbf{Z}_l^T) \\ \circ (\mathbf{e}_{q+1}^T \mathbf{Y}_k^T \mathbf{B} \mathbf{Y}_l \mathbf{e}_{r+1} \mathbf{L}_{\mathcal{G}}^{q+r})||_F^2 \end{aligned} \tag{4.8}$$

in terms of $\mathbf{\Gamma}$ and $\mathbf{B}$.

Next, for the term $f_2(\mathbf{\Gamma})$, we first observe that $f_2(\mathbf{\Gamma}) = \text{tr}(\mathbf{G}^T\mathbf{L}_\mathcal{G}\mathbf{G}) = \sum_{k=1}^{K} \mathbf{g}_k^T\mathbf{L}_\mathcal{G}\mathbf{g}_k$. Writing $\mathbf{L}_\mathcal{G} = \sum_{i=1}^{N} \lambda_\mathcal{G}(i)\, \mathbf{u}_i\mathbf{u}_i^T$ and using the equality $\mathbf{g}_k\mathbf{g}_k^T = \mathbf{Z}_k\mathbf{\Gamma}\mathbf{Z}_k^T$, we obtain the explicit form of the term $f_2(\mathbf{\Gamma})$ in (4.5) as

$$f_2(\mathbf{\Gamma}) = \text{tr}(\mathbf{G}^T\mathbf{L}_\mathcal{G}\mathbf{G}) = \sum_{k=1}^{K}\sum_{i=1}^{N} \lambda_\mathcal{G}(i)\, \mathbf{u}_i^T\mathbf{Z}_k\mathbf{\Gamma}\mathbf{Z}_k^T\mathbf{u}_i. \tag{4.9}$$

4.1.3 Complexity Analysis of the Algorithm

We next analyze the complexity of the method proposed in Algorithm 1 for learning LSGP models. The complexity of the preliminary step of finding the eigenvalue decomposition of $\mathbf{L}_\mathcal{G}$ is $O(N^3)$. The sample covariance estimation in Step-1 has complexity $O(N^2L)$. The most significant stage of the algorithm is Step-2, where we compute the $\mathbf{\Gamma}$ and $\mathbf{B}$ matrices by solving the optimization problem (4.5) based on semidefinite programming (SDP). The commonly used HKM algorithm can be taken as reference for the solution of SDP problems [60], whose complexity is $O(mn^3 + m^2n^2)$ with m and n respectively denoting the number of equality constraints and the number of variables. Hence, the complexity of the alternating stages of solving for $\mathbf{\Gamma}$ and $\mathbf{B}$ can be obtained as $O(\text{poly}(N)K^2)$ and $O(\text{poly}(K)Q^6)$ respectively, where poly$(\cdot)$ denotes at least cubic polynomial complexity. Next, the complexity of computing rank-1 decompositions for obtaining the model parameters in Step-3 is $O(N^3K^3)$ for $\mathbf{g}$ and $O(K^3Q^3)$ for $\mathbf{b}$. The evaluation of the polynomial functions in Step-4 requires $\Theta(NKQ)$ operations. Once the filter kernels $\mathbf{h}_k$ are found, Step-5 can be executed with a complexity of $O(N^2K)$. Finally, the estimation of the operator $\mathbf{H}$ from $\mathbf{M}$ in Step-6 has complexity $O(N^3)$. Hence, assuming that $K, Q \ll N$, the overall complexity of the algorithm can be reported as $O(\text{poly}(N)K^2)$.

Remark 2. *In the above analysis, the primary computational bottleneck of Algorithm 1 is seen to be Step-2, which has polynomial complexity in the number of nodes N. The high complexity in N stems mainly from the nonparametric formulation of the membership functions in our model, which results in N optimization variables to solve for, for each $\mathbf{g}_k$ vector. In applications involving large networks, the computational complexity of the algorithm can be alleviated through several strategies. For instance, the membership functions $\mathbf{g}_k$ can be formulated in a parametric form with*

a relatively small number of parameters, e.g., in terms of a linear combination of a small set of localized and smoothly varying graph signal prototypes, such as graph wavelets [53] and heat kernels [55]. This would significantly reduce the complexity of Step-2, while preserving the locality and smoothness properties of the membership functions over the graph. Another strategy would be to locally approximate the graph with a smaller subgraph and learn a simpler model on the small graph. This idea is elaborated in detail in Chapter 5.

4.1.4 Optimization Details

To solve the program defined in (4.5), the CVX package on MATLAB is utilized [61,62]. The problem in (4.5) consists of a Quadratic Program (QP) with semidefinite constraints, which with some effort can be turned into a Second Order Cone Program (SOCP) [63]. In this section, we will show how the optimization can be done without using the CVX package. Without loss of generality, assume $\mathbf{B}$ is kept fixed. Let γ be the vectorization of $\mathbf{\Gamma}$. Then the program in (4.5) can be written in vector form as

$$\begin{aligned} &\underset{\gamma}{\text{minimize}}\ \gamma^T \mathbf{Q} \gamma + \mathbf{b}^T \gamma \\ &\text{subject to } \mathbf{\Gamma} \in \mathbb{S}_+^{NK} \end{aligned} \tag{4.10}$$

for some $\mathbf{Q}, \mathbf{b}$ that make (4.10) and (4.5) equivalent. The solution of the problem in (4.10) can be achieved in two alternative ways. There are several works dedicated to the solution of Quadratic SDPs, that have quadratic optimization problems with semidefinite constraints [64, 65]. Another approach is to define a mapping to convert the QP to an SOCP. To define this first decompose $\mathbf{Q} = \mathbf{Q}_D^T \mathbf{Q}_D$, where $\mathbf{Q}_D \in \mathbb{R}^{\text{rank}(\mathbf{Q}) \times N^2 K^2}$. Such a decomposition can be made with SVD or QR decomposition [66]. Moreover, define the variable [63]

$$\mathbf{z} = \begin{bmatrix} 2\mathbf{Q}_D \gamma \\ \mathbf{z}_{r+1} \\ \mathbf{z}_{r+2} \end{bmatrix} \tag{4.11}$$

where $r = \text{rank}(\mathbf{Q})$. Then (4.10) becomes equivalent to

$$\begin{aligned} &\underset{\mathbf{z},\gamma}{\text{minimize}}\ \mathbf{z}_{r+2} + \mathbf{b}^T\gamma \\ \text{subject to } &\mathbf{\Gamma} \in \mathbb{S}_+^{NK},\ \mathbf{z}_{r+1} + \mathbf{z}_{r+2} = 2 \\ &\mathbf{z}_{1:r} = 2\mathbf{Q}_D\gamma,\ \mathbf{z}_{r+2} \geq ||z_{1:r+1}|| \end{aligned} \tag{4.12}$$

The problem in (4.12) can be solved with the SDPT3 [67] which has the ability to solve mixture of conic constraints using primal-dual interior point methods, whose updates are arising from Gauss-Newton iteration. We have demonstrated how the CVX reconstructs the problem inside [68]. Another approach is to write the QP as an SDP. Notice that, any problem in an SOCP form can be written as a SDP [69] with the following trick. First, convert the problem in (4.10) to the epigraph form, i.e.

$$\begin{aligned} &\underset{t,\gamma}{\text{minimize}}\ t \\ \text{subject to } &\mathbf{\Gamma} \in \mathbb{S}_+^{NK},\ t \geq \gamma^T\mathbf{Q}\gamma + \mathbf{b}^T\gamma. \end{aligned} \tag{4.13}$$

Then by the Schur complement test for semidefiniteness given in [70], the problem is converted into a SDP as shown in (4.14) [69].

$$\begin{aligned} &\underset{t,\gamma}{\text{minimize}}\ t \\ \text{subject to } &\mathbf{\Gamma} \in \mathbb{S}_+^{NK},\ \begin{bmatrix} \mathbf{I}_{\text{rank}(\mathbf{Q})} & \mathbf{Q}_D\gamma \\ \gamma^T\mathbf{Q}_D^T & t - b^T\gamma \end{bmatrix} \in \mathbb{S}_+^{\text{rank}(\mathbf{Q})} \end{aligned} \tag{4.14}$$

Typically, rank$(\mathbf{Q}) = N^2$, since $\mathbf{C_x}$ has N^2 entries; therefore, the problem has high complexity. (4.14) can be solved with existing methods that are used to solve linear SDPs in the literature [71], since now it is converted to a SDP problem.

Notice that the same conversion can be made for $\mathbf{B}$, in which case $\mathbf{Q}_D \in \mathbb{R}^{\text{rank}(\mathbf{Q})\times K^2Q^2}$, where rank$(\mathbf{Q})$ is typically K^2Q^2.

CHAPTER 5

LOCALLY APPROXIMATING LSGPS WITH WSS PROCESSES

In Section 4.1, we have proposed a method for learning an LSGP model from realizations of the process. The statistics of data sets collected on large networks are in practice likely to vary gradually throughout the network, which justifies the assumption of local stationarity. On the other hand, a common problem in graph signal processing is the potential complexity of learning graph signal models over a whole network as the network size increases, which is illustrated by the complexity analysis of Algorithm 1 as well. Motivated by these observations, in this chapter we explore a constructive approach for handling local stationarity in large graphs. Our approach is based on partitioning a given graph $\mathcal{G}$ into a set of K disjoint subgraphs $\{\mathcal{G}_k\}_{k=1}^K$. We consider the LSGP model in (3.2) and express the process $\mathbf{x}$ on the original graph $\mathcal{G}$ as $\mathbf{x} = \sum_{k=1}^K \mathbf{G}_k \mathbf{x}_k$, where each component process

$$\mathbf{x}_k = \mathbf{U}_{\mathcal{G}} h_k(\mathbf{\Lambda}_{\mathcal{G}}) \mathbf{U}_{\mathcal{G}}^T \mathbf{w}$$

is a WSS graph process. We then would like to partition $\mathcal{G}$ such that the process $\mathbf{x}$ can be approximated through only the component process $\mathbf{x}_k$ over each subgraph $\mathcal{G}_k$. The feasibility of such an approximation of course depends on the specific LSGP model at hand; in particular, the structure of the membership functions $\mathbf{g}_k$ and the characteristics of the kernels $\mathbf{h}_k$ of the component processes. In Section 5.1, we explore the conditions on $\mathbf{g}_k$ and $\mathbf{h}_k$ that permit accurate local approximations of $\mathbf{x}$ with WSS processes as above. We then study the covariance matrix $\mathbf{C}_{\mathbf{x}}$ of the process $\mathbf{x}$ and show that, under these conditions, the process $\mathbf{x}$ is weakly correlated across different subgraphs $\mathcal{G}_k$. Finally, these theoretical findings give rise to an algorithm in Section 5.2 for suitably partitioning a graph $\mathcal{G}$ based on the covariance of the LSGP $\mathbf{x}$ on $\mathcal{G}$, and locally approximating the LSGP with a WSS process $\mathbf{x}_k$ on each subgraph.

5.1 Covariance Analysis of LSGPs

We consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ with N nodes. Let $\{\mathcal{G}_k\}_{k=1}^K$ be disjoint subgraphs with vertex sets $\{\mathcal{V}_k\}_{k=1}^K$ such that $\bigsqcup_{k=1}^K \mathcal{V}_k = \mathcal{V}$. For each subgraph $\mathcal{G}_k$, let $\mathbf{S}_k \in \{0,1\}^{|\mathcal{V}_k| \times N}$ denote a binary selection matrix representing an inclusion map between $\mathcal{G}_k$ and $\mathcal{G}$ as defined in Section 3.1.2. We consider an LSGP $\mathbf{x} = \sum_{k=1}^K \mathbf{G}_k \mathbf{U}_\mathcal{G} h_k(\mathbf{\Lambda}_\mathcal{G}) \mathbf{U}_\mathcal{G}^T \mathbf{w}$ on graph $\mathcal{G}$ whose membership functions $\mathbf{g}_k$'s have the following property:

Assumption 1. *Each membership function $\mathbf{g}_k$ is localized over the subgraph $\mathcal{G}_k$ such that there exist constants $\delta, \mu, \gamma > 0$ with $|\mathbf{S}_k \mathbf{G}_m| \preccurlyeq \delta \mathbf{S}_k$ for $k \neq m$, and $\mu \mathbf{S}_k \preccurlyeq \mathbf{S}_k \mathbf{G}_k \preccurlyeq \gamma \mathbf{S}_k$ for $k = 1, \ldots, K$.*

According to the above assumption, each membership function $\mathbf{g}_k$ must be relatively strong on the corresponding subgraph $\mathcal{G}_k$ as lower bounded by the parameter μ, while it should take weaker values on the other subgraphs $\mathcal{G}_m$ as upper bounded by the parameter δ. The parameter γ represents an upper bound that prevents $\mathbf{g}_k$ from taking unbounded values.

We first wish to determine how well the process $\mathbf{x}$ can be approximated by the component processes $\mathbf{x}_k$, which we characterize in terms of their second-order statistics. Let

$$\mathbf{C}_{\mathbf{x}_k \mathbf{x}_m} = E[\mathbf{x}_k \mathbf{x}_m^T]$$

denote the cross-covariance matrix between the component process $\mathbf{x}_k$ and $\mathbf{x}_m$. In the next main result, we provide a bound on the deviation between $\mathbf{C}_\mathbf{x}$ and the cross-covariances $\mathbf{C}_{\mathbf{x}_k \mathbf{x}_m}$ of the component processes.

Theorem 2. *Let Assumption 1 hold for the LSGP $\mathbf{x}$. Then for all $k, m \in \{1, \ldots, K\}$, the cross-covariances $\mathbf{C}_{\mathbf{x}_k \mathbf{x}_m}$ of the components of $\mathbf{x}$ approximate the overall covariance $\mathbf{C}_\mathbf{x}$ according to the following bound:*

$$\begin{aligned} &|\mathbf{S}_k \mathbf{G}_k^\dagger \mathbf{C}_\mathbf{x} (\mathbf{G}_m^\dagger)^T \mathbf{S}_m^T - \mathbf{S}_k \mathbf{C}_{\mathbf{x}_k \mathbf{x}_m} \mathbf{S}_m^T| \\ &\preccurlyeq \left(2(K-1)\frac{\delta}{\mu} + (K-1)^2 \left(\frac{\delta}{\mu}\right)^2 \right) \mathbf{1}_{|\mathcal{V}_k| \times |\mathcal{V}_m|} \end{aligned} \tag{5.1}$$

Before proceeding, let us introduce the following lemmas which we are going to use in the proof.

Lemma 2. *For any matrices* $\mathbf{U}, \mathbf{A}, \mathbf{V}$ *of compatible size such that* $\mathbf{U}\mathbf{A}\mathbf{V}^T \in \mathbb{R}^{M\times N}$*, the inequality* $|\mathbf{U}\mathbf{A}\mathbf{V}^T| \preccurlyeq |\mathbf{U}||\mathbf{A}||\mathbf{V}^T|$ *is satisfied.*

Proof. Let us denote the n-th rows of $\mathbf{U}$ and $\mathbf{V}$ respectively as $\mathbf{u}_n^T$ and $\mathbf{v}_n^T$. We then have

$$\mathbf{U}\mathbf{A}\mathbf{V}^T = \begin{bmatrix} \mathbf{u}_1^T\mathbf{A}\mathbf{v}_1 & \mathbf{u}_1^T\mathbf{A}\mathbf{v}_2 & \cdots & \mathbf{u}_1^T\mathbf{A}\mathbf{v}_N \\ \mathbf{u}_2^T\mathbf{A}\mathbf{v}_1 & \mathbf{u}_2^T\mathbf{A}\mathbf{v}_2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_M^T\mathbf{A}\mathbf{v}_1 & \cdots & \cdots & \mathbf{u}_M^T\mathbf{A}\mathbf{v}_N \end{bmatrix}.$$

For any (i, j) index pair, we obtain

$$\begin{aligned} |\mathbf{u}_i^T\mathbf{A}\mathbf{v}_j| &= \left|\sum_k\sum_l \mathbf{A}(k,l)\mathbf{u}_i^T(k)\mathbf{v}_j(l)\right| \\ &\leq \sum_k\sum_l |\mathbf{A}(k,l)|\,|\mathbf{u}_i^T(k)|\,|\mathbf{v}_j(l)| = |\mathbf{u}_i^T|\,|\mathbf{A}|\,|\mathbf{v}_j|. \end{aligned}$$

It follows that $|\mathbf{U}\mathbf{A}\mathbf{V}^T| \preccurlyeq |\mathbf{U}||\mathbf{A}||\mathbf{V}^T|$. □

Lemma 3. *Let* $\mathbf{U}, \mathbf{A}, \mathbf{B}, \mathbf{V}$ *be real matrices of compatible size such that* $\mathbf{U}\mathbf{A}\mathbf{V}^T \in \mathbb{R}^{M\times N}$ *and* $\mathbf{U}\mathbf{B}\mathbf{V}^T \in \mathbb{R}^{M\times N}$*. Assume that* $\mathbf{U}, \mathbf{V} \succcurlyeq 0$ *and* $\mathbf{B} \preccurlyeq \mathbf{A}$*. Then* $\mathbf{U}\mathbf{B}\mathbf{V}^T \preccurlyeq \mathbf{U}\mathbf{A}\mathbf{V}^T$.

Proof. Fixing any index pair (i, j), and using the same notation for row vectors as in the proof of Lemma 2, we have

$$\begin{aligned} \mathbf{u}_i^T\mathbf{B}\mathbf{v}_j &= \sum_k\sum_l \mathbf{B}(k,l)\mathbf{u}_i^T(k)\mathbf{v}_j^T(l) \\ &\leq \sum_k\sum_l \mathbf{A}(k,l)\mathbf{u}_i^T(k)\mathbf{v}_j^T(l) = \mathbf{u}_i^T\mathbf{A}\mathbf{v}_j \end{aligned}$$

since $\mathbf{B} \preccurlyeq \mathbf{A}$. □

Proof. We can now prove Theorem 2. The cross-covariance matrix of the component processes $\mathbf{x}_k$ and $\mathbf{x}_m$ is given by

$$\mathbf{C}_{\mathbf{x}_k\mathbf{x}_m} = \mathbf{U}_\mathcal{G} h_k(\mathbf{\Lambda}_\mathcal{G}) h_m(\mathbf{\Lambda}_\mathcal{G}) \mathbf{U}_\mathcal{G}^T. \tag{5.2}$$

The element-wise magnitude of the matrix $\mathbf{C}_{\mathbf{x}_k\mathbf{x}_m}$ can be bounded as

$$\begin{aligned}|\mathbf{C}_{\mathbf{x}_k\mathbf{x}_m}| &= \left|\sum_{i=1}^{N}\mathbf{h}_k(i)\mathbf{h}_m(i)\mathbf{u}_i\mathbf{u}_i^T\right| \preccurlyeq \sum_{i=1}^{N}|\mathbf{h}_k(i)\mathbf{h}_m(i)\mathbf{u}_i\mathbf{u}_i^T| \\ &\preccurlyeq \sum_{i=1}^{N}|\mathbf{h}_k(i)\mathbf{h}_m(i)|\mathbf{1}_{N\times N} \preccurlyeq \frac{1}{2}\sum_{i=1}^{N}\left(|\mathbf{h}_k^2(i)|+|\mathbf{h}_m^2(i)|\right)\mathbf{1}_{N\times N} \\ &\preccurlyeq \mathbf{1}_{N\times N}\end{aligned} \tag{5.3}$$

where the second and the fourth inequalities follow respectively from the fact that the vectors $\mathbf{u}_i$ and $\mathbf{h}_k$ are unit-norm.

Next, we write the covariance matrix $\mathbf{C}_\mathbf{x}$ of the process $\mathbf{x}$ as a weighted average of the cross-covariances $\mathbf{C}_{\mathbf{x}_k\mathbf{x}_m}$ as

$$\mathbf{C}_\mathbf{x} = \sum_{k=1}^{K}\sum_{m=1}^{K}\mathbf{G}_k\mathbf{C}_{\mathbf{x}_k\mathbf{x}_m}\mathbf{G}_m^T. \tag{5.4}$$

We can then bound the deviation between $\mathbf{C}_{\mathbf{x}_k\mathbf{x}_m}$ and the restriction of $\mathbf{C}_\mathbf{x}$ to the subgraphs $\mathcal{G}_k$ and $\mathcal{G}_m$ as

$$\begin{aligned}&\left|\mathbf{S}_k\mathbf{G}_k^\dagger\mathbf{C}_\mathbf{x}(\mathbf{G}_m^\dagger)^T\mathbf{S}_m^T - \mathbf{S}_k\mathbf{C}_{\mathbf{x}_k\mathbf{x}_m}\mathbf{S}_m^T\right| \\ &= \left|\sum_{(i,j)\neq(k,m)}\mathbf{S}_k\mathbf{G}_k^\dagger\mathbf{G}_i\mathbf{C}_{\mathbf{x}_i\mathbf{x}_j}\mathbf{G}_j^T(\mathbf{G}_m^\dagger)^T\mathbf{S}_m^T\right| \\ &\preccurlyeq \sum_{(i,j)\neq(k,m)}|\mathbf{S}_k\mathbf{G}_k^\dagger\mathbf{G}_i||\mathbf{C}_{\mathbf{x}_i\mathbf{x}_j}||\mathbf{G}_j^T(\mathbf{G}_m^\dagger)^T\mathbf{S}_m^T| \\ &\preccurlyeq \sum_{(i,j)\neq(k,m)}|\mathbf{S}_k\mathbf{G}_k^\dagger\mathbf{G}_i|\,\mathbf{1}_{N\times N}\,|\mathbf{G}_j^T(\mathbf{G}_m^\dagger)^T\mathbf{S}_m^T|\end{aligned} \tag{5.5}$$

where the first and the second inequalities are due to Lemmas 2 and 3, respectively. In order to bound the expression in (5.5) in terms of δ and μ, we next examine the product $|\mathbf{S}_k\mathbf{G}_k^\dagger\mathbf{G}_i|$ for the cases $i \neq k$ and $i = k$. Due to Assumption 1, we have $|\mathbf{S}_k\mathbf{G}_k^\dagger\mathbf{G}_i| \preccurlyeq \frac{1}{\mu}\mathbf{S}_k|\mathbf{G}_i| \preccurlyeq \frac{\delta}{\mu}\mathbf{S}_k$ for $i \neq k$; and $|\mathbf{S}_k\mathbf{G}_k^\dagger\mathbf{G}_i| \preccurlyeq \mathbf{S}_k$ for $i = k$. Using these

inequalities in (5.5), we get

$$\begin{aligned}
&\left|\mathbf{S}_k\mathbf{G}_k^\dagger\mathbf{C}_{\mathbf{x}}(\mathbf{G}_m^\dagger)^T\mathbf{S}_m^T - \mathbf{S}_k\mathbf{C}_{\mathbf{x}_k\mathbf{x}_m}\mathbf{S}_m^T\right| \\
&\preccurlyeq \sum_{(i,j)\neq(k,m)} |\mathbf{S}_k\mathbf{G}_k^\dagger\mathbf{G}_i|\,\mathbf{1}_{N\times N}\,|\mathbf{G}_j^T(\mathbf{G}_m^\dagger)^T\mathbf{S}_m^T| \\
&= \sum_{i\neq k, j=m} |\mathbf{S}_k\mathbf{G}_k^\dagger\mathbf{G}_i|\,\mathbf{1}_{N\times N}\,|\mathbf{G}_j^T(\mathbf{G}_m^\dagger)^T\mathbf{S}_m^T| \\
&\sum_{i=k, j\neq m} |\mathbf{S}_k\mathbf{G}_k^\dagger\mathbf{G}_i|\,\mathbf{1}_{N\times N}\,|\mathbf{G}_j^T(\mathbf{G}_m^\dagger)^T\mathbf{S}_m^T| \\
&+ \sum_{i\neq k, j\neq m} |\mathbf{S}_k\mathbf{G}_k^\dagger\mathbf{G}_i|\,\mathbf{1}_{N\times N}\,|\mathbf{G}_j^T(\mathbf{G}_m^\dagger)^T\mathbf{S}_m^T| \\
&\preccurlyeq \left(2(K-1)\frac{\delta}{\mu} + (K-1)^2\left(\frac{\delta}{\mu}\right)^2\right)\mathbf{S}_k\mathbf{1}_{N\times N}\mathbf{S}_m^T \\
&= \left(2(K-1)\frac{\delta}{\mu} + (K-1)^2\left(\frac{\delta}{\mu}\right)^2\right)\mathbf{1}_{|\mathcal{V}_k|\times|\mathcal{V}_m|}
\end{aligned} \tag{5.6}$$

which concludes the proof. □

Theorem 2 compares the covariance $\mathbf{C}_{\mathbf{x}}$ of the process $\mathbf{x}$ (normalized by the inverse membership functions for appropriate scaling) with the cross-covariance $\mathbf{C}_{\mathbf{x}_k\mathbf{x}_m}$ of the component processes, when locally restricted to the nodes on the subgraphs $\mathcal{G}_k$ and $\mathcal{G}_m$. Note that by choosing $k = m$, the statement of the theorem pertains to the approximation of $\mathbf{C}_{\mathbf{x}}$ by the covariance $\mathbf{C}_{\mathbf{x}_k}$ of the component process $\mathbf{x}_k$ on the subgraph $\mathcal{G}_k$. The theorem implies that as the ratio δ/μ decreases, which is a measure of how much the supports of the memberships $\mathbf{g}_k$'s are restricted to the subgraphs $\mathcal{G}_k$'s, the covariance of $\mathbf{x}$ can be more accurately approximated by the covariances of $\mathbf{x}_k$'s. This result brings about the possibility of identifying suitable subgraphs $\mathcal{G}_k$'s such that the LSGP $\mathbf{x}$ can be approximated with the WSS process $\mathbf{x}_k$ when restricted to the subgraph $\mathcal{G}_k$. However, in order to accurately simplify the problem of computing an LSGP model $\mathbf{x}$ to the computation of independent WSS process models $\mathbf{x}_k$ on smaller subgraphs, the processes $\{\mathbf{x}_k\}$ must be weakly correlated with each other; i.e., the covariance matrix $\mathbf{C}_{\mathbf{x}}$ must have negligible entries over its off-diagonal blocks corresponding to $\mathbf{C}_{\mathbf{x}_k\mathbf{x}_m}$ for $k \neq m$. In order to establish this condition, in addition to Assumption 1, the kernels $\{\mathbf{h}_k\}$ must also be sufficiently different from each other, which we characterize via their spectral separation in the following assumption:

Assumption 2. *For all $k, m \in \{1, 2, \ldots, K\}$ with $k \neq m$, the spectral supports of the kernels $\mathbf{h}_k, \mathbf{h}_m$ are separated from each other such that $\sum_{i=1}^{N} |\mathbf{h}_k(i)\, \mathbf{h}_m(i)| \leq \epsilon$.*

The parameter $\epsilon \geq 0$ in Assumption 2 is thus a spectral separation parameter ensuring the incoherence of the kernels $\mathbf{h}_k$. The dissimilarities of the membership functions $\mathbf{g}_k$'s as well as the separation of the kernels $\mathbf{h}_k$ imposed by Assumptions 1 and 2 then guarantee an upper bound on the process cross-covariance values across different subgraphs, which is stated in the next main result.

Theorem 3. *Let Assumptions 1 and 2 hold. Then, the average squared cross-covariance of $\mathbf{x}$ across different subgraphs $\mathcal{G}_k$, $\mathcal{G}_m$ is upper bounded as*

$$\begin{aligned}\frac{1}{\gamma^4 N^2} \sum_{k=1}^{K} \sum_{\substack{m=1 \\ m \neq k}}^{K} \sum_{(i,j) \in \mathcal{V}_k \times \mathcal{V}_m} |\mathbf{C}_{\mathbf{x}}(i,j)|^2 \leq \frac{2}{N} K(K-1)\epsilon^2 \\ + \frac{2}{N^2} \left| \bigcup_{k=1}^{K} \bigcup_{\substack{m=1 \\ m \neq k}}^{K} \mathcal{V}_k \times \mathcal{V}_m \right| \left(2(K-1)\frac{\delta}{\mu} + (K-1)^2 \left(\frac{\delta}{\mu}\right)^2 \right)^2 .\end{aligned} \tag{5.7}$$

Proof. We begin by observing that for any $a, b \in \mathbb{R}$, we have

$$|a+b|^2 \leq |a|^2 + |b|^2 + 2|ab| \leq 2\left(|a|^2 + |b|^2\right). \tag{5.8}$$

We can bound the average squared cross-covariance as

$$
\begin{aligned}
&\frac{1}{\gamma^4}\sum_{k=1}^{K}\sum_{\substack{m=1\\m\neq k}}^{K}\sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_m}\left|\mathbf{C}_{\mathbf{x}}(i,j)\right|^2\\
&\leq\sum_{k=1}^{K}\sum_{\substack{m=1\\m\neq k}}^{K}\sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_m}\mathbf{g}_k(i)^{-2}\left|\mathbf{C}_{\mathbf{x}}(i,j)\right|^2\mathbf{g}_m(j)^{-2}\\
&\leq 2\sum_{k=1}^{K}\sum_{\substack{m=1\\m\neq k}}^{K}\sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_m}\left|\mathbf{g}_k(i)^{-1}\mathbf{C}_{\mathbf{x}}(i,j)\mathbf{g}_m(j)^{-1}\right.\\
&\qquad\left.-\,\mathbf{C}_{\mathbf{x}_k\mathbf{x}_m}(i,j)\right|^2+2\sum_{k=1}^{K}\sum_{\substack{m=1\\m\neq k}}^{K}\sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_m}\left|\mathbf{C}_{\mathbf{x}_k\mathbf{x}_m}(i,j)\right|^2\\
&\leq 2\sum_{k=1}^{K}\sum_{\substack{m=1\\m\neq k}}^{K}\sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_m}\left(2(K-1)\frac{\delta}{\mu}+(K-1)^2\left(\frac{\delta}{\mu}\right)^2\right)^2\\
&+2\sum_{k=1}^{K}\sum_{\substack{m=1\\m\neq k}}^{K}\sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_m}\left|\mathbf{C}_{\mathbf{x}_k\mathbf{x}_m}(i,j)\right|^2
\end{aligned}
\tag{5.9}
$$

where the last inequality is due to Theorem 2. We proceed by upper bounding the

cross-covariance sum as

$$
\begin{aligned}
&\sum_{k=1}^{K}\sum_{\substack{m=1\\m\neq k}}^{K}\sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_m}|\mathbf{C}_{\mathbf{x}_k\mathbf{x}_m}(i,j)|^2\\
&=\sum_{k=1}^{K}\sum_{\substack{m=1\\m\neq k}}^{K}\sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_m}\left|\sum_{l=1}^{N}\mathbf{h}_k(l)\mathbf{h}_m(l)\mathbf{u}_l(i)\mathbf{u}_l(j)\right|^2\\
&\leq\sum_{k=1}^{K}\sum_{\substack{m=1\\m\neq k}}^{K}\sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_m}\left(\sum_{l=1}^{N}\mathbf{h}_k(l)^2\mathbf{h}_m(l)^2\right)\\
&\qquad\qquad\cdot\left(\sum_{n=1}^{N}\mathbf{u}_n(i)^2\mathbf{u}_n(j)^2\right) \qquad (5.10)\\
&=\sum_{l=1}^{N}\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{\substack{m=1\\m\neq k}}^{K}\mathbf{h}_k(l)^2\mathbf{h}_m(l)^2\sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_m}\mathbf{u}_n(i)^2\mathbf{u}_n(j)^2\\
&=\sum_{l=1}^{N}\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{\substack{m=1\\m\neq k}}^{K}\mathbf{h}_k(l)^2\mathbf{h}_m(l)^2\sum_{i\in\mathcal{V}_k}\mathbf{u}_n(i)^2\sum_{j\in\mathcal{V}_m}\mathbf{u}_n(j)^2\\
&\leq N\sum_{l=1}^{N}\sum_{k=1}^{K}\sum_{\substack{m=1\\m\neq k}}^{K}\mathbf{h}_k(l)^2\mathbf{h}_m(l)^2\leq NK(K-1)\epsilon^2
\end{aligned}
$$

where the last inequality is due to Assumption 2. Using this result in (5.9), we get the bound stated in the theorem

$$
\begin{aligned}
&\frac{1}{\gamma^4N^2}\sum_{k=1}^{K}\sum_{\substack{m=1\\m\neq k}}^{K}\sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_m}|\mathbf{C}_{\mathbf{x}}(i,j)|^2\\
&\leq\frac{2}{N}K(K-1)\epsilon^2\\
&+\frac{2}{N^2}\sum_{k=1}^{K}\sum_{\substack{m=1\\m\neq k}}^{K}\sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_m}\left(2(K-1)\frac{\delta}{\mu}+(K-1)^2\left(\frac{\delta}{\mu}\right)^2\right)^2 \qquad (5.11)\\
&=\frac{2}{N}K(K-1)\epsilon^2\\
&+\frac{2}{N^2}\left|\bigcup_{k=1}^{K}\bigcup_{\substack{m=1\\m\neq k}}^{K}\mathcal{V}_k\times\mathcal{V}_m\right|\left(2(K-1)\frac{\delta}{\mu}+(K-1)^2\left(\frac{\delta}{\mu}\right)^2\right)^2.
\end{aligned}
$$

□

In the theorem, the first term in the cross-covariance upper bound decreases with ϵ; hence, when the kernels $\mathbf{h}_k$ have lesser frequency content in common, the correlation between the process values on different subgraphs $\mathcal{G}_k$, $\mathcal{G}_m$ weakens. Next, the second term in the upper bound reflects the effect of the localization of the membership functions on the process cross-covariance. As each membership function $\mathbf{g}_k$ attains better localization on the corresponding subgraph $\mathcal{G}_k$, the ratio δ/μ decreases due to Assumption 1, weakening the covariance of the process $\mathbf{x}$ across different subgraphs.

While Theorem 3 guarantees an upper bound on the cross-covariance of $\mathbf{x}$ across different subgraphs $\mathcal{G}_k$, $\mathcal{G}_m$, in order for this bound to be meaningful, it should be assessed relatively to the covariance of $\mathbf{x}$ on each subgraph $\mathcal{G}_k$. Therefore, our next aim is to get a lower bound for the average strength of the process covariance $\mathbf{C}_{\mathbf{x}}$ over the individual subgraphs $\mathcal{G}_k$. In order to ensure such a lower bound, the process $\mathbf{x}$ must change sufficiently slowly on each subgraph $\mathcal{G}_k$, which requires a restriction on the bandwidths of the kernels $\mathbf{h}_k$ as follows:

Assumption 3. *The kernels $\mathbf{h}_k$ are band-limited such that $\mathbf{h}_k(i) = 0$ for $i > \kappa_C$ for all $k \in \{1, 2, \ldots, K\}$, where κ_C is a cutoff parameter with $\kappa_C \in \{1, 2, \ldots, N\}$.*

Before proceeding to our next main result, we also define the following parameters related to the topology of $\mathcal{G}$ and the process characteristics:

Definition 3. *Let $D(i, j)$ denote the unweighted geodesic distance between two vertices $i, j \in \mathcal{V}$ given by*

$$D(i,j) \triangleq \min\{n : \exists\, (l_0, l_1, \ldots, l_n) \text{ such that } l_k \sim l_{k+1}$$
$$\text{for } k = 0, \ldots, n-1; l_k \in \{1, \ldots, N\};\ l_0 = i,\ l_n = j\}.$$

Also let $w_{min} \triangleq \min_{i \sim j} \mathbf{W}(i,j)$ denote the minimum edge weight on $\mathcal{G}$, let $T_n = \sum_{i \sim j} \mathbf{W}(i,j)(\mathbf{u}_n(i) - \mathbf{u}_n(j))^2$ denote the total variation of $\mathbf{u}_n$ on $\mathcal{G}$, and let $\sigma_k^2 \triangleq \sum_{i \in \mathcal{V}_k} \mathbf{C}_{\mathbf{x}_k}(i,i)$ represent the total variance of the process $\mathbf{x}_k$ on the subgraph $\mathcal{G}_k$.

We can now present our last result, which gives a lower bound on the average magnitude of the process covariance on the individual subgraphs.

Theorem 4. *Let Assumptions 1 and 3 hold. Then the average magnitude of the co-*

variance of the process $\mathbf{x}$ *on the individual subgraphs* $\{\mathcal{G}_k\}$ *is lower bounded as*

$$\begin{aligned}\frac{1}{N^2\mu^2}\sum_{k=1}^{K}\sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_k}|\mathbf{C}_{\mathbf{x}}(i,j)| &\geq \frac{1}{N^2}\sum_{k=1}^{K}|\mathcal{V}_k|\sigma_k^2 \\ -\frac{1}{2\,N^2\,w_{min}}\sum_{k=1}^{K}\sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_k}D^2(i,j)\sum_{n=1}^{\kappa_C}T_n & \\ -\frac{1}{N^2}\left|\bigsqcup_{k=1}^{K}\mathcal{V}_k\times\mathcal{V}_k\right|\left(2(K-1)\frac{\delta}{\mu}+(K-1)^2\left(\frac{\delta}{\mu}\right)^2\right).&\end{aligned} \tag{5.12}$$

We first present the following lemma, which will be useful for the proof of Theorem 4.

Lemma 4. *The following inequality holds for all edges* $(i,j)\in\mathcal{E}$ *and all* $n\in\{1,\ldots,N\}$.

$$\frac{T_n}{w_{min}}\geq(\mathbf{u}_n(i)-\mathbf{u}_n(j))^2 \tag{5.13}$$

Also, for all vertex pairs $(i,j)\in\mathcal{V}\times\mathcal{V}$,

$$D^2(i,j)\frac{T_n}{w_{min}}\geq(\mathbf{u}_n(i)-\mathbf{u}_n(j))^2. \tag{5.14}$$

Proof. We have

$$\begin{aligned}T_n&=\sum_{i\sim j}\mathbf{W}(i,j)(\mathbf{u}_n(i)-\mathbf{u}_n(j))^2\\ &\geq w_{min}\sum_{i\sim j}(\mathbf{u}_n(i)-\mathbf{u}_n(j))^2\\ &\geq w_{min}\max_{i\sim j}(\mathbf{u}_n(i)-\mathbf{u}_n(j))^2.\end{aligned} \tag{5.15}$$

Hence, (5.13) is proved. For showing (5.14), consider a simple path $(i,l_1,\ldots,l_{q-1},j)$ of length $q=D(i,j)$ between nodes i and j. Then,

$$\begin{aligned}&D(i,j)\sqrt{\frac{T_n}{w_{min}}}\\ &\geq|\mathbf{u}_n(i)-\mathbf{u}_n(l_1)|+\cdots+|\mathbf{u}_n(l_{q-1})-\mathbf{u}_n(j)|\\ &\geq|\mathbf{u}_n(i)-\mathbf{u}_n(j)|.\end{aligned} \tag{5.16}$$

Taking the square of both sides, we get the inequality in (5.14). □

We can now prove Theorem 4.

Proof. We first obtain an expression for the total covariances of the component processes on their corresponding subgraphs as

$$\begin{aligned}
&\sum_{k=1}^{K} \sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_k} \mathbf{C}_{\mathbf{x}_k}(i,j) \\
&= \sum_{k=1}^{K} \sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_k} \sum_{n=1}^{\kappa_C} \mathbf{h}_k(n)^2\, \mathbf{u}_n(i)\, \mathbf{u}_n(j) \\
&= \frac{1}{2} \sum_{k=1}^{K} \sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_k} \sum_{n=1}^{\kappa_C} \mathbf{h}_k(n)^2 \left(\mathbf{u}_n(i)^2 + \mathbf{u}_n(j)^2\right) \\
&- \frac{1}{2} \sum_{k=1}^{K} \sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_k} \sum_{n=1}^{\kappa_C} \mathbf{h}_k(n)^2 \left(\mathbf{u}_n(i) - \mathbf{u}_n(j)\right)^2
\end{aligned} \tag{5.17}$$

where the first equality is due to Assumption 3. We first obtain the following relation

$$\begin{aligned}
&\sum_{k=1}^{K} \sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_k} \sum_{n=1}^{\kappa_C} \mathbf{h}_k(n)^2 \mathbf{u}_n(i)^2 \\
&= \sum_{k=1}^{K} \sum_{i\in\mathcal{V}_k} \sum_{n=1}^{\kappa_C} |\mathcal{V}_k|\, \mathbf{h}_k(n)^2 \mathbf{u}_n(i)^2 \\
&= \sum_{k=1}^{K} \sum_{i\in\mathcal{V}_k} |\mathcal{V}_k| \mathbf{C}_{\mathbf{x}_k}(i,i) = \sum_{k=1}^{K} |\mathcal{V}_k|\, \sigma_k^2.
\end{aligned} \tag{5.18}$$

Next, from Lemma 4, we get

$$\begin{aligned}
-\sum_{n=1}^{\kappa_C} \mathbf{h}_k(n)^2 (\mathbf{u}_n(i) - \mathbf{u}_n(j))^2 &\geq -\sum_{n=1}^{\kappa_C} (\mathbf{u}_n(i) - \mathbf{u}_n(j))^2 \\
&\geq -\frac{D^2(i,j)}{w_{min}} \sum_{n=1}^{\kappa_C} T_n.
\end{aligned} \tag{5.19}$$

Using (5.18) and (5.19) in (5.17), we obtain

$$\begin{aligned}
&\sum_{k=1}^{K} \sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_k} \mathbf{C}_{\mathbf{x}_k}(i,j) \geq \sum_{k=1}^{K} |\mathcal{V}_k| \sigma_k^2 \\
&- \frac{1}{2w_{min}} \sum_{k=1}^{K} \sum_{(i,j)\in\mathcal{V}_k\times\mathcal{V}_k} D^2(i,j) \sum_{n=1}^{\kappa_C} T_n.
\end{aligned} \tag{5.20}$$

Finally, from Assumption 1 it follows that

$$\begin{aligned}\frac{1}{\mu^2}|\mathbf{C}_{\mathbf{x}}(i,j)| &\geq |\mathbf{G}_k^{\dagger}(i,i)\mathbf{C}_{\mathbf{x}}(i,j)\mathbf{G}_k^{\dagger}(j,j)| \\ &\geq \mathbf{G}_k^{\dagger}(i,i)\mathbf{C}_{\mathbf{x}}(i,j)\mathbf{G}_k^{\dagger}(j,j) \\ &\geq \mathbf{C}_{\mathbf{x}_k}(i,j) - |\mathbf{G}_k^{\dagger}(i,i)\mathbf{C}_{\mathbf{x}}(i,j)\mathbf{G}_k^{\dagger}(j,j) - \mathbf{C}_{\mathbf{x}_k}(i,j)|.\end{aligned} \tag{5.21}$$

Using this result together with the bound in Theorem 2, we get the inequality stated in Theorem 4.

□

In Theorem 4, the first term in the right hand side of (5.12) sets a reference value for the average covariance magnitude of $\mathbf{x}$ when restricted to the subgraphs $\{\mathcal{G}_k\}$, assuming that the number K of subgraphs is increased proportionally to growing graph size N. The average covariance magnitude has limited deviation from this reference value if the second and the third terms have restricted magnitudes. The second term improves as the bandwidth parameter κ_C of the kernels decreases, while it can also be controlled by regulating the diameters of the subgraphs $\{\mathcal{G}_k\}$'s [72]. The third term is bounded by the localization ratio δ/μ of the membership functions as in Theorem 3.

5.2 Algorithm

The analysis in Sec. 5.1 shows that, under certain assumptions, the second-order statistics of an LSGP model can be approximated by those of individual locally-defined WSS graph processes. In this section, inspired by these results, we propose an algorithm for partitioning a given graph $\mathcal{G}$ into smaller subgraphs, such that an LSGP $\mathbf{x}$ defined on $\mathcal{G}$ can be approximated by an individual WSS process $\mathbf{x}_k$ on each subgraph $\mathcal{G}_k$. Note that the WSS process model $\mathbf{x}_k$ on each subgraph can be obtained by computing an LSGP model with a single component by setting $K = 1$.

Our partitioning algorithm relies on the results in Theorems 3 and 4 in particular. These results state that if the LSGP model $\mathbf{x}$ admits a local approximation, the cross-covariance of $\mathbf{x}$ across different subgraphs must be relatively weak, while ensuring a

lower bound on the covariance of the process on each individual subgraph. Assuming that an initial estimate $\hat{\mathbf{C}}_{\mathbf{x}}$ of the covariance matrix is available, we thus propose to inspect the covariance matrix $\hat{\mathbf{C}}_{\mathbf{x}}$ along with the graph topology $\mathcal{G}$ in order to identify a set of subgraphs $\{\mathcal{G}_k\}_{k=1}^K$, such that the weak entries in $\hat{\mathbf{C}}_{\mathbf{x}}$ are associated with between-subgraph cross-covariance values, and the strong entries in $\hat{\mathbf{C}}_{\mathbf{x}}$ correspond to within-subgraph covariance values.

In order to determine the subgraphs $\{\mathcal{G}_k\}$, we first use the covariance values in $\hat{\mathbf{C}}_{\mathbf{x}}$ for defining a distance function $\rho_{\mathbf{x}} : \mathcal{E} \to \mathbb{R}^+$ on the edges $\mathcal{E}$ of the graph $\mathcal{G}$. The distance function $\rho_{\mathbf{x}}(i,j) = d(\hat{\mathbf{C}}_{\mathbf{x}}(i,j))$ is computed through a continuous and even kernel $d : \mathbb{R} \to \mathbb{R}^+$ that is strictly decreasing on $\mathbb{R}^+ \cup \{0\}$. A suitable choice for $d(x)$ is the Gaussian function $\exp(-x^2/\theta)$, where the parameter θ adjusts the mapping between the covariance values and the corresponding distances. Once the distance function $\rho_{\mathbf{x}}(i,j)$ is obtained, we employ a graph partitioning algorithm $\mathcal{P}(\mathcal{G}, \rho_{\mathbf{x}}, K)$ that partitions the graph $\mathcal{G}$ into K disjoint subgraphs $\{\mathcal{G}_k\}_{k=1}^K$ by cutting the edges (i,j) associated with high distance values $\rho_{\mathbf{x}}(i,j)$ and retaining those with low distances. Many alternatives exist in the literature for the choice of the partitioning algorithm $\mathcal{P}$; an example method can be found in the study [73], where the edges in $\mathcal{G}$ are progressively removed in a geometry-dependent manner based on the Ricci curvature induced by the distance $\rho_{\mathbf{x}}$. The resulting graph partitioning procedure is denoted as `partitionGraph` in Algorithm 2. Once the subgraphs $\{\mathcal{G}_k\}_{k=1}^K$ are determined, we construct their selection matrices $\{\mathbf{S}_k\}_{k=1}^K$, and restrict the realizations $\mathbf{x}^l$ of the original process $\mathbf{x}$ on $\mathcal{G}$ to each subgraph as $\mathbf{x}_k^l = \mathbf{S}_k \mathbf{x}^l$. One can then compute a WSS graph process on each subgraph $\mathcal{G}_k$ through the method described in Algorithm 1 by setting $K = 1$ in the LSGP model. The overall procedure is outlined in Algorithm 2.

Algorithm 2 Local Approximation of LSGPs

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, realizations $\{\mathbf{x}^l\}_{l=1}^L$, number of subgraphs K, distance function d, graph partitioning method $\mathcal{P}$

Output: Subgraphs $\{\mathcal{G}_k\}$, an individual LSGP model $(\mathbf{g}_k, \mathbf{h}_k)$ on each subgraph $\mathcal{G}_k$.

Estimate initial covariance $\hat{\mathbf{C}}_\mathbf{x}$ from $\{\mathbf{x}^l\}_{l=1}^L$
$\{\mathcal{G}_k\}_{k=1}^K \leftarrow$ `partitionGraph` $\left(\mathcal{G}, \hat{\mathbf{C}}_\mathbf{x}, \mathcal{P}, d\right)$
for $k = 1, 2, \dots K$
    Construct selection matrix $\mathbf{S}_k$ for subgraph $\mathcal{G}_k$
    for $l = 1, 2, \dots L$
        $\mathbf{x}_k^l = \mathbf{S}_k \mathbf{x}^l$
    end
    $(\mathbf{g}, \mathbf{h}) \leftarrow$ `learnLSGP`$(\mathcal{G}_k, \{\mathbf{x}_k^l\}_{l=1}^L, 1)$
    $\mathbf{g}_k = \mathbf{g};\ \mathbf{h}_k = \mathbf{h}$
end

procedure `partitionGraph`$(\mathcal{G}, \hat{\mathbf{C}}_\mathbf{x}, \mathcal{P}, d)$
$\rho_\mathbf{x} : \mathcal{E} \to \mathbb{R}^+$
for $(i, j) \in \mathcal{E}$
    $\rho_\mathbf{x}(i, j) = d\left(\hat{\mathbf{C}}_\mathbf{x}(i, j)\right)$
end
$\{\mathcal{G}_k\}_{k=1}^K \leftarrow \mathcal{P}(\mathcal{G}, \rho_\mathbf{x}, K)$
return $\{\mathcal{G}_k\}_{k=1}^K$

CHAPTER 6

EXPERIMENTAL RESULTS

In this chapter, we evaluate the performance of the proposed algorithms on synthetic and real world datasets.

6.1 Performance Analysis of Algorithm 1

We first study the sensitivity of the LSGP learning method proposed in Algorithm 1.

6.1.1 Sensitivity to noise level

We construct a synthetic 5-NN graph with $N = 36$ nodes from a 2D grid perturbed with additive white Gaussian noise. The original grid consists of 36 points sampled regularly from the rectangle $[0, 1]^2$ with 6 evenly distributed points in each direction. The noise standard deviation in each direction x, y is selected as $\sigma_x = \sigma_y = \frac{0.6}{\sqrt{2}}$. A graph is constructed from this noisy point cloud with the edge weights determined with the Gaussian kernel, $W(i, j) = \exp(-\frac{||x_i - x_j||^2}{\theta})$, where $\theta = 0.9$. We then generate realizations of an LSGP $\mathbf{x}$ whose vertex-frequency spectrum is given in Fig. 3.1, with parameters $K = 3, Q = 4$ on this graph according to the model (3.2). The realizations are corrupted with additive white Gaussian noise at different signal-to-noise ratio (SNR) levels.

We study two problems. In the first problem, we initially compute a sample covariance estimate $\hat{\mathbf{C}}_{\mathbf{x}}$ of the process from all available realizations, then learn a model based on $\hat{\mathbf{C}}_{\mathbf{x}}$ with Algorithm 1, and finally obtain an estimate $\mathbf{C}^*_{\mathbf{x}} = \mathbf{H}\mathbf{H}^T$ of the covariance according to the learnt model.

Before learning the covariance matrix according to (4.5), we normalize the sample covariance matrix, because while estimating unknowns with LMMSE, scale information does not provide a factor. Therefore, we assess the covariance performance by comparing to the normalized true covariance matrix, and look at the Frobenius norm of the error, which is called Covariance Discrepancy (CD). In this case, note that CD in (6.1) also measures how much the learnt covariance matrix deviates from the unit ball as well as the deviation from correlation pattern.

$$\text{CD} = ||\mathbf{C}_\mathbf{x} - \mathbf{C}^*_\mathbf{x}||_F \tag{6.1}$$

where $C_\mathbf{x}$ is normalized true covariance matrix.

In the second problem, we address a scenario where 10000 realizations $\mathbf{x}^l$ of the process are given, with half of the values in the realizations being missing. We obtain the covariance matrix $\mathbf{C}^*_\mathbf{x}$ with the proposed LSGP method as above, and then obtain the LMMSE estimates of the missing values in each realization $\mathbf{x}^l$ as

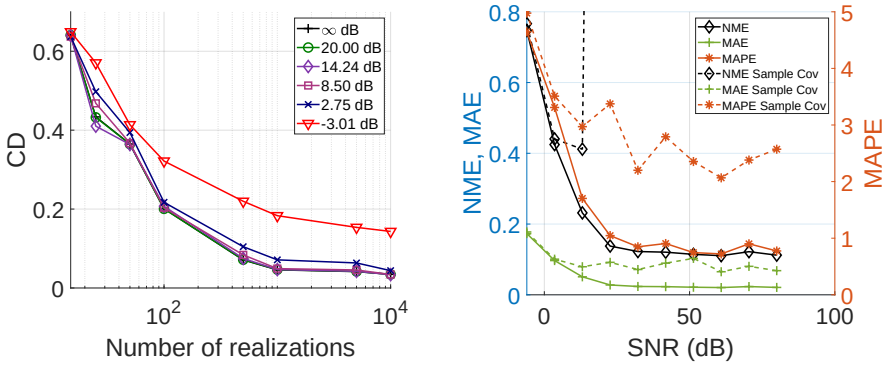
$$\hat{\mathbf{z}}^l = (\mathbf{C}^*_{\mathbf{zy}})^l \left((\mathbf{C}^*_\mathbf{y})^l\right)^{-1} \mathbf{y}^l \tag{6.2}$$

where the vectors $\mathbf{y}^l$ and $\mathbf{z}^l$ respectively contain the available and the initially missing entries of each realization $\mathbf{x}^l$, and $\hat{\mathbf{z}}^l$ is the estimate of $\mathbf{z}^l$. The matrix $(\mathbf{C}^*_{\mathbf{zy}})^l$ denotes the estimated cross-covariance of $\mathbf{z}^l$ and $\mathbf{y}^l$, and $(\mathbf{C}^*_\mathbf{y})^l$ is the estimated covariance of $\mathbf{y}^l$, which can be obtained by extracting the corresponding entries of $\mathbf{C}^*_\mathbf{x}$ for each realization $\mathbf{x}^l$. Defining a concatenated vector $\mathbf{z}$ that consists of the missing values $\mathbf{z}^l$ in all realizations and its estimate $\hat{\mathbf{z}}$, we evaluate the estimation error with respect to the normalized mean error (NME), the mean absolute error (MAE), and the mean absolute percentage error (MAPE) metrics, as shown in (6.3a), (6.3b), (6.3c), respectively, where $L_\mathbf{z}$ denotes the length of $\mathbf{z}$.

$$\text{NME} = \frac{||\mathbf{z} - \hat{\mathbf{z}}||_2}{||\mathbf{z}||_2} \tag{6.3a}$$

$$\text{MAE} = \frac{||\mathbf{z} - \hat{\mathbf{z}}||_1}{L_\mathbf{z}} \tag{6.3b}$$

$$\text{MAPE} = \frac{1}{L_\mathbf{z}} \sum_{i=1}^{L} \frac{|\mathbf{z}(i) - \hat{\mathbf{z}}(i)|}{|\mathbf{z}(i)|} \tag{6.3c}$$



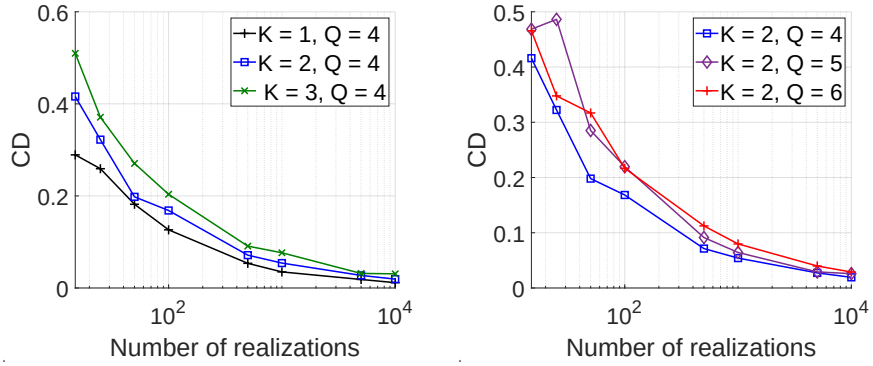
(a) (b)

Figure 6.1: (a) Variation of the covariance discrepancy with respect to the number of realizations. (b) Variation of the estimation errors with the SNR.

The covariance discrepancies and the estimation errors in the two problems are plotted in Fig. 6.1(a) and Fig. 6.1(b), respectively. The dashed curves in Fig. 6.1(b) are provided as reference, showing the LMMSE estimation errors obtained with the initial sample covariance matrix $\hat{\mathbf{C}}_{\mathbf{x}}$ given as input to Algorithm 1. In both figures, the estimation performance improves with increasing SNR as expected. In Fig. 6.1(a), the covariance discrepancy remains quite close to the ideal infinite SNR case and approaches 0 with increasing number of realizations until the SNR drops down to -3 dB. In these results, the CD of the sample covariance estimate $\hat{\mathbf{C}}_{\mathbf{x}}$ is consistently above that of the algorithm output $\mathbf{C}_{\mathbf{x}}^{*}$, with the gap reaching around 0.05 at -3 dB and around 0.04 at infinite SNR, which are not included in the plots for visual clarity. Despite the seemingly minor difference in the CD values of $\hat{\mathbf{C}}_{\mathbf{x}}$ and $\mathbf{C}_{\mathbf{x}}^{*}$, the estimation performance of $\mathbf{C}_{\mathbf{x}}^{*}$ is in fact significantly better than $\hat{\mathbf{C}}_{\mathbf{x}}$ in the interpolation problem in Fig. 6.1(b), which shows the efficacy of the proposed method for improving the initial estimate of the process statistics.

6.1.2 Effect of model complexity

We next examine the effect of the model order parameters K (number of process components) and Q (polynomial order) on the estimation performance of LSGP. A synthetic 7-NN graph $\mathcal{G}$ with $N = 36$ nodes is formed as in Section 6.1.1 by com-



(a) (b)

Figure 6.2: Variation of the covariance discrepancy with the model complexity

bining K subgraphs $\{\mathcal{G}_k\}_{k=1}^K$ each of which consists of $36/K$ nodes. The component processes $\mathbf{x}_k$ are blended in $\mathbf{x}$ by setting the membership functions $\mathbf{g}_k$ to 1 within each subgraph $\mathcal{G}_k$ and to 0.1 outside $\mathcal{G}_k$.

The variation of the covariance discrepancy is plotted in Fig. 6.2(a) for variable K by fixing $Q = 4$, and for variable Q in Fig. 6.2(b) by fixing $K = 2$. As the model complexity increases, the number of realizations required to attain a target covariance discrepancy level increases in both cases as expected. The CD converges to 0 with increasing number of realizations in all cases, which confirms that Algorithm 1 recovers the true process model. The algorithm performance is more sensitive to the K parameter than Q, which is expected since increasing K by 1 increases the model dimension by N.

6.1.3 Sensitivity to regularization parameters

We lastly investigate the effect of the weight parameters μ_1, μ_2, μ_3 in the optimization problem (4.5) on the algorithm performance. We conduct these experiments on real data sets instead of synthetic data sets, since a synthetically generated graph process fits the assumed signal model perfectly, which may affect the behavior of the algorithm concerning the regularization parameters. We experiment on the COVID-19 and the Molène data sets described in Section 6.3. The ratio of missing observations

Table 6.1: Variation of the NME and MAPE with μ_1

μ_1	0	10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}
Dataset	COVID-19							
NME	0.8091	0.8091	0.8091	0.8091	**0.8091**	0.8092	0.8130	0.8210
MAPE	2.1454	**2.1454**	2.1455	2.1459	2.1503	2.2075	2.4162	2.5038
Dataset	Molène							
NME	0.4470	0.4470	0.4470	0.4470	0.4470	0.4470	0.4465	**0.4438**
MAPE	1.0257	**1.0257**	1.0257	1.0257	1.0257	1.0258	1.0267	1.0369

is fixed to 80%. The K and Q parameters have been chosen by majority voting on validation data, which has resulted in $K = Q = 3$ on the COVID-19 dataset and the same parameters have been used on the Molène dataset. Once the model is learnt with the proposed method, the LMMSE estimates of the missing observations are obtained as in (6.2), and the NME and MAPE metrics for the LMMSE estimates are reported in Table 6.1 for varying μ_1 values and in Table 6.2 for varying μ_2, μ_3 combinations. The non-tested weight parameters are fixed to 0 in each experiment, in order to focus on the effect of the tested ones.

In Table 6.1, the performance of the algorithm is seen to be stable with respect to the variations in μ_1 over a rather large interval, which controls the smoothness of the membership functions. The COVID-19 data set favors slightly lower μ_1 values compared to Molène, thus imposing the smoothness less strictly. The NME and MAPE values in Molène are more robust to the increase in μ_1, meaning that the objective function will not change much when μ_1 is increased, hence allowing identification with smoother $\mathbf{g}$. This is an expected result because if we use the stationarity measure convention in [6], the stationarity of the COVID-19 data set is given around 0.76 and Molène is given around 0.94. Thus, there is a possibility that the membership functions of Molène are close to the first Laplacian eigenvector [74], while COVID-19 allows more abrupt changes in the membership functions than Molène. In the COVID-19 data set, from 0 to 10^{-5}, the NME value of LMMSE is approximately constant with the minimum occuring at 10^{-5}. However, this is not true for MAPE where the maximum between 0 and 10^{-5} resides at 10^{-5}. Hence, although $\mu_1 = 10^{-5}$

Table 6.2: Variation of the NME and MAPE with μ_2 and μ_3

μ_2 \ μ_3	0	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
Metric	NME (COVID-19)							
0	0.8091	0.8091	0.8091	0.8091	0.8091	0.8091	9.5685	0.8091
10^{-7}	0.8091	0.8091	0.8091	0.8091	0.8091	0.8091	0.8093	0.8096
10^{-6}	0.8091	0.8091	0.8091	0.8091	0.8091	0.8091	0.8089	0.8118
10^{-5}	0.8091	0.8091	0.8091	0.8091	0.8093	0.8097	0.8089	0.8112
10^{-4}	0.8090	0.8090	0.8090	0.8090	0.8095	0.8101	0.8104	0.8328
10^{-3}	0.8096	0.8096	0.8096	0.8101	0.8098	0.8116	0.8133	0.9125
10^{-2}	**0.8084**	**0.8084**	**0.8084**	**0.8085**	0.8091	0.8126	0.9107	0.9392
10^{-1}	0.8105	0.8093	0.8095	0.8113	0.8126	0.9116	0.9533	0.9950
Metric	MAPE (COVID-19)							
0	2.1454	2.1454	2.1454	2.1454	2.1454	2.1454	2.6417	2.1456
10^{-7}	2.1455	2.1455	2.1455	2.1455	2.1456	2.1458	2.1298	2.2688
10^{-6}	2.1460	2.1460	2.1460	2.1460	2.1469	2.1492	2.1489	2.2752
10^{-5}	2.1508	2.1508	2.1508	2.1512	2.1530	2.1384	2.1703	2.3944
10^{-4}	2.1959	2.1959	2.1960	2.2140	2.2051	2.2234	2.2491	2.1158
10^{-3}	2.3035	2.3034	2.3029	2.2797	2.2344	2.3254	2.3276	1.6488
10^{-2}	2.2116	2.2118	2.2136	2.2237	2.2798	2.4312	1.7388	1.8048
10^{-1}	2.1697	2.1727	2.2483	2.2779	2.4244	1.7243	1.8460	**1.0557**

leads to smaller NME, the MAPE score seems more unstable as μ_1 increases. The MAPE value is optimal at $\mu_1 = 10^{-8}$, hence a possible choice is to pick a value between 10^{-8} and 10^{-6} to attain a trade-off between the two scores.

Next, the results in Table 6.2 show that relatively small (μ_2, μ_3) values lead to smaller NME. Recalling that these parameters impose the low-rank constraints in (4.5), the decrease in the MAPE values for increasing (μ_2, μ_3) is misleading as the algorithm tends to compute a zero process model under too heavy regularization. This can also be seen from the NME being 0.995 at $(0.1, 0.1)$. The NME value is lowest at $(\mu_2, \mu_3) = (10^{-2}, 10^{-6})$ and since from our discussion as higher values of (μ_2, μ_3)

leads to a zero process model, it is safe to say that lower values of the regularization parameters are better. In order to find a trade-off between the two scores, it is better to pick μ_2 between $10^{-4} - 10^{-5}$ and μ_3 between $0 - 10^{-5}$.

6.2 Performance Analysis of Algorithm 2

We next verify the validity of our theoretical findings in Chapter 5 by conducting a performance analysis of Algorithm 2. We construct a synthetic graph similar to the one in Section 6.1.2 consisting of $K = 5$ subgraphs and a total of $N = 300$ nodes. The subgraphs $\mathcal{G}_k$ are built with a 7-NN connectivity pattern within themselves and combined with each other via extra edges. The membership functions $\mathbf{g}_k$ to component processes are chosen as in Section 6.1.2. The spectral kernels $\mathbf{h}_k$ are set by shifting and scaling a compactly supported bump function according to the intended spectral separation, which is defined as $h(\lambda) = \exp((\lambda^{2n} - 1)^{-1})$ for $\lambda \in (-1, 1)$ and 0 elsewhere.

We study the problem of graph partitioning for locally approximating LSGPs and examine how the membership ratio δ/μ and the spectral separation parameter ϵ defined in Chapter 5 influence the performance of Algorithm 2. In each instance of the experiment, LSGPs with the investigated δ/μ and ϵ parameters are generated where $\epsilon = 0$ and $\delta/\mu = 0.3$ are kept constant respectively in each experiment. Algorithm 2 is provided the true covariance matrix $\mathbf{C_x}$ generated from the true LSGP parameters $\mathbf{g}, \mathbf{h}$ according to (4.1). The distance function $d(x)$ in Section 5.2 is chosen as the Gaussian function given by $d(x) = \exp\left(\frac{-x^2}{\theta}\right)$, where θ is a suitable value that can separate different $\hat{\mathbf{C}}_\mathbf{x}$ entries. The partitioning algorithm is selected as Ricci curvature with surgery [73] as mentioned, where the surgery is done via maximizing the modularity of the negative log weight distribution of the output of the Ricci curvature expansion algorithm where the maximum number of partitions returned by modularity is set to the true value $K = 5$ [75]. The subgraphs $\{\hat{\mathcal{G}}_k\}_{k=1}^K$ returned by the algorithm are compared to the true subgraphs $\{\mathcal{G}_k\}$.

We evaluate the agreement between the true and the estimated subgraphs using the

Table 6.3: Variation of the NMI with $\frac{\delta}{\mu}$ and ϵ

Membership ratio		**Spectral separation**	
$\frac{\delta}{\mu}$	NMI	ϵ	NMI
0	0.9764	0.1	0.9822
0.13	0.979	0.2	0.9853
0.27	0.965	0.3	0.9740
0.4	0.962	0.4	0.9488
0.53	0.918	0.5	0.9559
0.67	0.917	0.6	0.9489
0.8	0.853	0.7	0.9306

normalized mutual information (NMI) measure defined as[1]

$$\text{NMI} = \frac{1}{\max(H(\mathcal{P}_{\mathcal{V}}), H(\mathcal{P}_{\hat{\mathcal{V}}}))} \sum_{k,m} p(\mathcal{V}_k, \hat{\mathcal{V}}_m) \log_2 \frac{p(\mathcal{V}_k, \hat{\mathcal{V}}_m)}{p(\mathcal{V}_k)p(\hat{\mathcal{V}}_m)} \tag{6.4}$$

where $\mathcal{P}_{\mathcal{V}} = \{\mathcal{V}_1, \ldots, \mathcal{V}_K\}$ is the true partitioning of the vertices and $\mathcal{P}_{\hat{\mathcal{V}}} = \{\hat{\mathcal{V}}_1, \ldots, \hat{\mathcal{V}}_K\}$ denotes its estimate. In (6.4), $p(\mathcal{V}_k)$ and $p(\mathcal{V}_k, \mathcal{V}_m)$ represent the probability that a vertex chosen uniformly at random lies in $\mathcal{V}_k$ and in $\mathcal{V}_k \cap \mathcal{V}_m$, respectively. Also, $H(\mathcal{P}_{\mathcal{V}}) = -\sum_k p(\mathcal{V}_k) \log_2 p(\mathcal{V}_k)$ denotes the entropy of a partitioning. A higher value of the NMI thus indicates a stronger agreement between the two partitions.

Table 6.3 shows the variation of the NMI with the parameters δ/μ and ϵ. As the δ/μ ratio increases, the dominance of each membership function on the corresponding subgraph weakens, leading to a decrease in the NMI. This is coherent with the findings of Theorems 3 and 4, stating that δ/μ must be low for ensuring weak between-subgraph and strong within-subgraph covariances. Similarly, the NMI decreases as ϵ increases, which reduces the separation between the kernels and affects the partitioning performance, in coherence with Theorem 3. One might notice that, although the general variation of the NMI values is coherent with the expected trend, there may be small fluctuations in the NMI, which may be due to the following reasons: First, the bounds proved in Theorem 3 and 4 are worst case bounds for the average entry of the

[1] Among the various definitions of NMI, we adopt the definition in which normalization is performed using the maximum of the entropies of the partitionings.

covariance matrix, while the individual covariance values may change with respect to the graph topology, the choice of the component processes etc. The second reason is that partitioning algorithms are combinatorial rather than continuous, which would may lead to momentary abrupt changes in the NMI, as $\frac{\delta}{\mu}$ or ϵ values vary.

6.3 Comparative Experiments on Real Data Sets

In this section, we evaluate our signal estimation performance on three real data sets:

COVID-19 pandemic data set. The COVID-19 data set consists of the number of daily new COVID-19 cases in $N = 37$ European countries of highest populations between February 15, 2020 and July 5, 2021 [76]. A 4-NN graph is constructed by considering each country as a graph node. Edge weights are determined with a Gaussian kernel based on a hybrid distance measure that combines geographical distances and numbers of flights accessed via [77]. The daily new cases are normalized by country populations and smoothed out with a moving average filter over one week, which results in a graph process with 506 realizations.

Molène weather data set. This data set consists of hourly temperature measurements taken in $N = 37$ measurement stations in the Brittany region of France in January 2014 [4]. The graph is constructed with a 5-NN topology by considering each station as a graph node, with Gaussian edge weights based on the geographical coordinates (lattitude, longitude) of the stations. Experiments are done on 744 zero mean graph signals.

NOAA weather data set. The NOAA data set contains hourly temperature measurements for one year taken in weather stations across the US averaged over the years 1981-2010 [78]. We construct a 7-NN graph from $N = 246$ weather stations with Gaussian edge weights. The experiments are done on 8760 graph signals.

6.3.1 Comparative performance evaluation of the LSGP algorithm

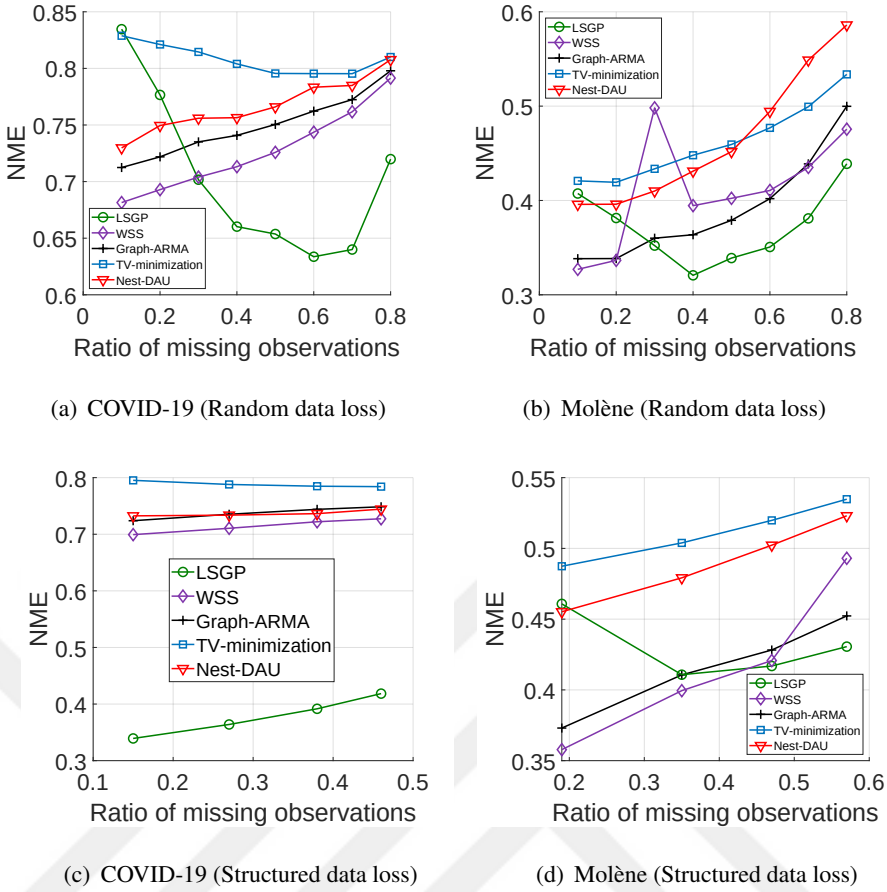
We first study a signal interpolation problem on the Molène and COVID-19 data sets by considering two scenarios:

- (Random data loss) Missing observations of the graph signals occur at nodes selected uniformly at random.

- (Structured data loss) Missing observations occur at particular regions of the graph over a local clique of neighboring graph nodes.

When testing the proposed method (LSGP), each graph signal is treated as the realization of a locally stationary graph process, an LSGP model is learnt with Algorithm 1, and LMMSE estimates of the missing observations are computed as in (6.2). The proposed LSGP model is compared to two other stochastic graph process models; namely, wide sense stationary graph processes (WSS) [6] and graph ARMA processes (Graph-ARMA) [5]. We also include two reference non-stochastic graph signal interpolation approaches in our comparisons, based on the total variation regularization of graph signals (TV-minimization) [79], and the deep algorithm unrolling method (Nest-DAU) recently proposed in [80]. All algorithms employing the process covariance matrix have been provided the sample covariance estimate as input. For algorithms requiring a covariance matrix, a sample covariance matrix has been estimated on the available observations of the realizations. However, this might create a covariance matrix with negative eigenvalues. For LSGP, the sample covariance matrix is provided as input to the algorithm as it is. For the WSS approach, the input covariance matrix is converted to a positive semi definite matrix by setting the negative eigenvalues to zero after diagonalization. An MA process has been learnt with the Graph-ARMA method. Algorithm hyperparameters are determined with validation for all methods that require parameter tuning. For all algorithms, the validation procedure is repeated for each realization and missing number of observations. For the LSGP method, first the K, Q parameters, and then using these, the μ_1, μ_2, μ_3 parameters have been validated. For each realization K is swept between 1 and 3 and Q is swept between 1 and 5.

The variation of the estimation errors of the methods with the ratio of missing observations is shown in Fig. 6.3-6.5 with respect to the NME, MAE, and MAPE metrics. The signal estimation performance of the proposed LSGP method is seen to be competitive with the other methods, often outperforming them especially at middle-to-high missing observation ratios. In most instances, the stochastic process based meth-



(a) COVID-19 (Random data loss)

(b) Molène (Random data loss)

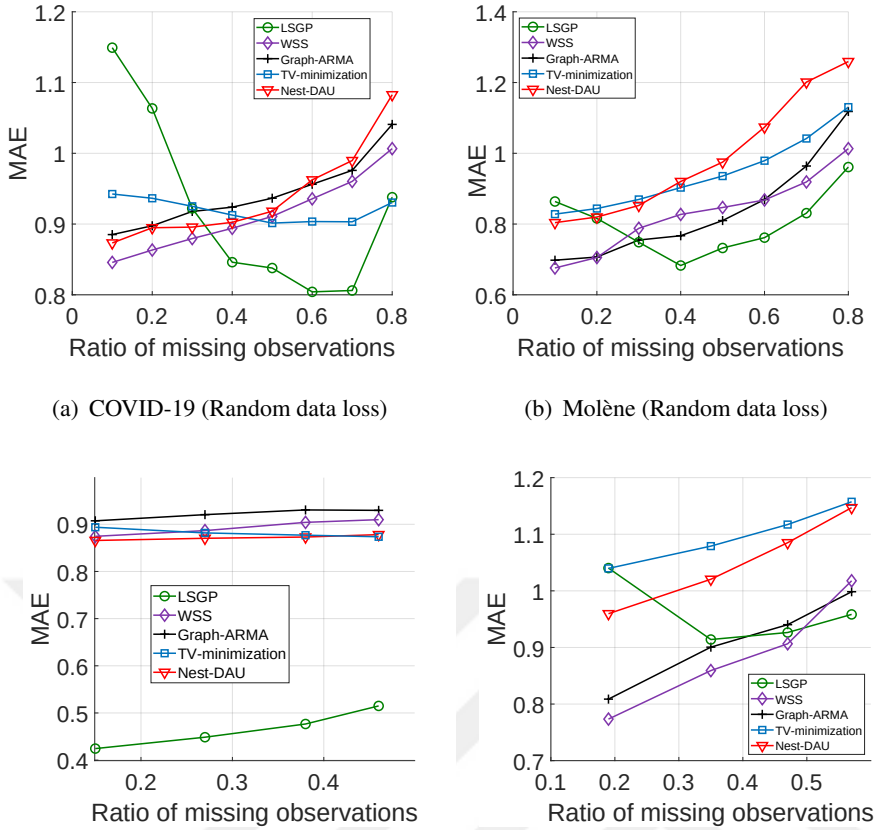
(c) COVID-19 (Structured data loss)

(d) Molène (Structured data loss)

Figure 6.3: NME of compared algorithms on COVID-19 and Molène data sets

ods LSGP, WSS, and Graph-ARMA provide smaller error than the non-stochastic TV-minimization and Nest-DAU methods.

The results on the random data loss experiments suggest that the error of the LSGP algorithm tends to be rather high at small ratios of missing observations. This somewhat unexpected behavior may be explained with the reasoning that, at small ratios of missing observations, the choice of the algorithm hyperparameters tailored to the available data samples may not generalize well to the missing data, which is small in proportion. This effect is reduced as the ratio of missing observations increases and thus the balance between the available and missing samples improves. This phenomenon is observed at a milder level on the Molène data set than COVID-19. The variation of the membership functions is smoother on the Molène data set [74], leading to a higher similarity between the missing and the available observations.



(a) COVID-19 (Random data loss)

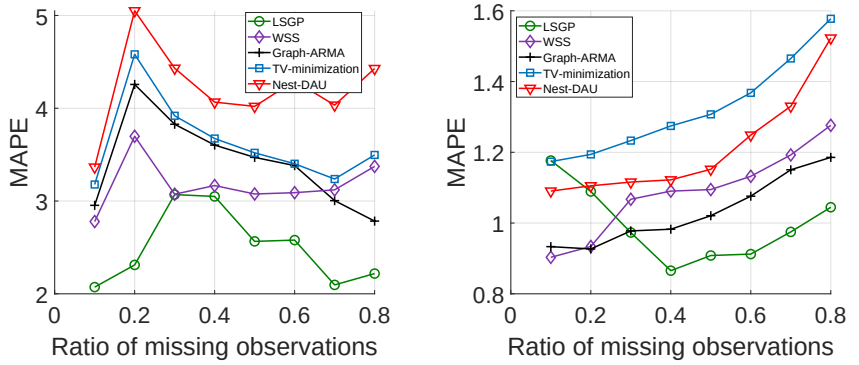
(b) Molène (Random data data loss)

(c) COVID-19 (Structured data loss)

(d) Molène (Structured data loss)

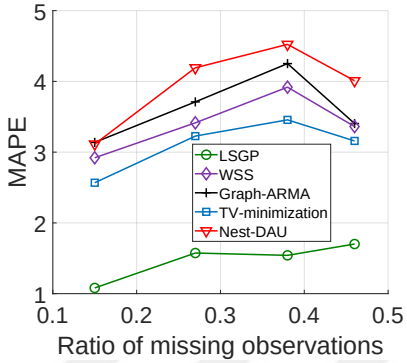
Figure 6.4: MAE of compared algorithms on COVID-19 and Molène data sets

In the structured data loss experiment, for Molène and COVID-19, each realization has randomly selected neighborhoods whose number is between 1 and 4, and the task is again to estimate the missing samples. The proposed LSGP method significantly outperforms the other methods in comparison and achieves error values smaller than half of those of the other methods. While the error of LSGP in structured data loss experiments is smaller than that in random data loss on the COVID-19 data set, the two settings result in comparable error values on the Molène data set. One possible explanation for this might be the smoothness of the associated membership functions. Since the graph signals vary more smoothly over the graph in the Molène data set, when a neighborhood is missing for a realization the loss of information within that neighborhood can be compensated via the information available on the nearby neighborhoods. Therefore, the results obtained in the structured and random data loss

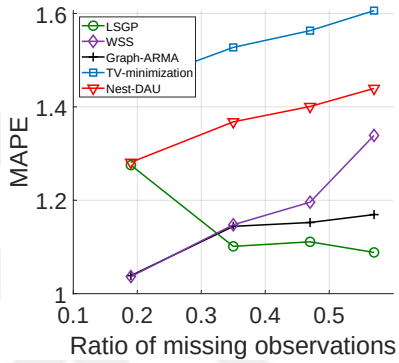


(a) COVID-19 (Random data loss)

(b) Molène (Random data loss)



(c) COVID-19 (Structured data loss)



(d) Molène (Structured data loss)

Figure 6.5: MAPE of compared algorithms on COVID-19 and Molène data sets

scenarios are similar on the Molène data set. In COVID-19, this is not the case, however. Graph signals have a rougher variation on the graph in the COVID-19 data set, therefore, there is a gap between the NME, MAE and MAPE values of the random and structured data loss scenarios.

6.3.2 Local approximation of LSGPs

We finally study the performance of locally approximating LSGPs with smaller processes in problems where one needs to analyze data acquired on large network topologies. We experiment on the NOAA weather data set [78]. While the NOAA graph with $N = 246$ nodes is not a typical representative of a "large network", we have chosen this data set deliberately so that we can conveniently test signal estimation

algorithms on both the whole graph and its separate subgraphs in order to study how partitioning affects the estimation performance.

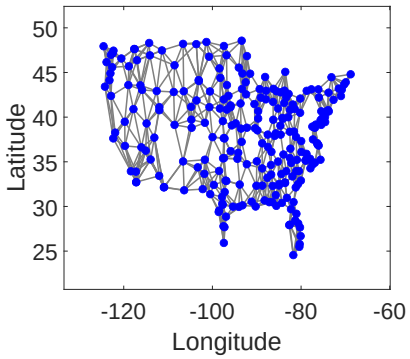
We first partition the NOAA graph using Algorithm 2 where we set the number of subgraphs as $K = 7$. The original graph and the computed subgraphs are shown in Fig. 6.6(a) and Fig. 6.6(b). For comparison, we also present the climate map of the US [81] in Fig. 6.6(c), which interestingly suggests that the estimated graph partition boundaries mostly align with the climate transition zones often marked by geographical elements such as mountain ranges. A close examination of the results reveals that the partitions in the west are approximately divided by the Rocky mountain range [82], whose topography is responsible for local temperature fluctuations in its proximity and the partitions in the east are divided by the Appalachian mountain range and partly the Grenville orogenic belt [83] extending towards the south. We select the missing observations uniformly at random and consider two signal estimation settings for the compared methods: In the first setting we learn distinct models on separate subgraphs (Separate), while in the second setting we learn a single model on the whole graph (Whole). As in Section 6.3.1 the algorithm hyperparameters are selected by validation on each realization and this time the K parameter for LSGP is kept constant at 1 and Q is swept in the range $1 - 10$ instead of the smaller range used previously. The algorithms are tested in both the separate subgraphs and the whole graph settings and compared with respect to the NME, MAE and MAPE metrics in Fig. 6.6(d)-6.6(f). The estimation of the signals on the whole graph often provides higher accuracy than on separate subgraphs, as expected. An interesting exception to this occurs in Fig. 6.6(e), where estimation on separate subgraphs results in smaller MAE, since the error in separate modeling concentrates sparsely along subgraph boundaries and thus has relatively small ℓ_1-norm. An illustration of this phenomenon where we plot the MAE value obtained at each node for 50% and 60% missing observation ratios is given in Fig. 6.7. It is seen that more error is accumulated towards the boundaries where the inner regions have mostly small error. Regarding the difference between the whole graph and separate subgraphs settings, the method with the smallest performance gap between the two settings is the TV-minimization method, whose natural tendency to compute piecewise smooth signals is coherent with separate modeling. The performance of LSGP is competitive with that of TV-

minimization and often better than the other methods on separate subgraphs, offering a promising solution in this setting. We also compare the time complexities of the methods in Table 6.4 by reporting their average runtimes[2]. The runtimes of stochastic process methods differ by a factor of around 4 between the separate subgraph and whole graph settings, where the LSGP method has not been tested in the latter one due to its complexity. This situation illustrates a common challenge faced by many graph signal processing algorithms, whose complexities are typically around $O(N^3)$, making them impractical to use in a straightforward way as the graph size N grows. The experiments in this section have aimed to provide some insight for handling the scalability issue in large graphs via the local modeling and processing of graph signals on suitably identified neighborhoods.

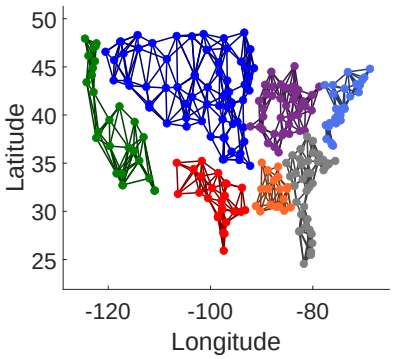
Table 6.4: Runtimes of the compared methods (in seconds) on the NOAA data set at 50% missing observation ratio

Algorithm	**Separate subgraphs**	**Whole graph**
LSGP	62.39	-
WSS	5.14	20.30
Graph-ARMA	4.90	18.34
TV-minimization	338.84	408.20
Nest-DAU	6145.51	1015.38

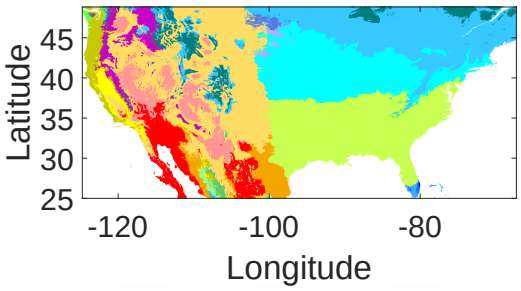
[2] The runtimes are measured on a laptop computer with 16 GB DDR5 RAM, and 3.2 GHz CPU. The Nest-DAU method has a higher runtime in the separate setting since the validation procedure has resulted in a larger number of layers.



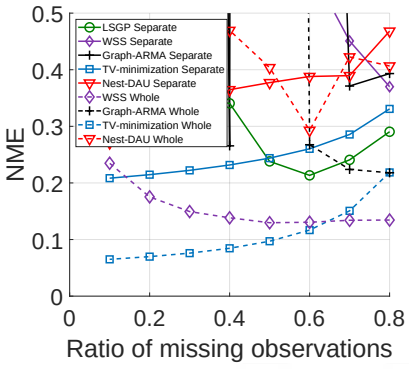
(a) Original NOAA graph



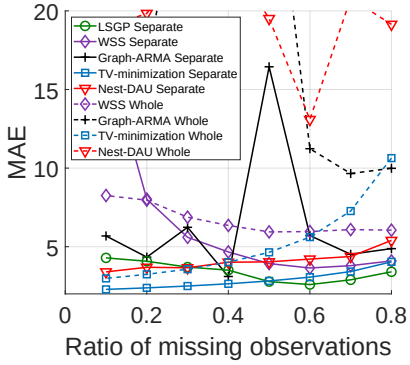
(b) Partitioning of the NOAA graph



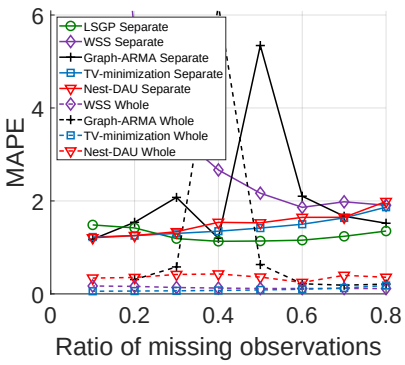
(c) Köppen-Geiger climate regions of US



(d) NME for NOAA data set

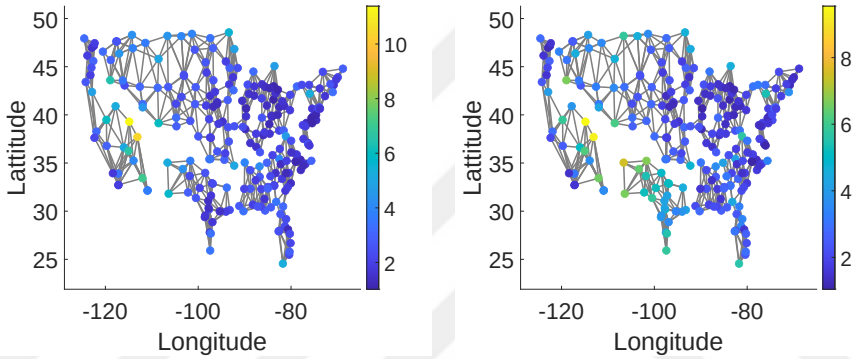


(e) MAE for NOAA data set



(f) MAPE for NOAA data set

Figure 6.6: Results obtained on the NOAA data set



(a) MAE value at each node for missing observation ratio 60%

(b) MAE value at each node for missing observation ratio 50%

Figure 6.7: MAE values at each node for selected missing observation ratios

CHAPTER 7

CONCLUSIONS

In this thesis, we have proposed a graph signal model called LSGP, which is composed of several component process models, hence, allows the modelling of graph data having different characteristics over different local regions on the graph. This is achieved by defining membership functions for each component model, which represent the validity of each component on each node. We have then studied the main properties of the proposed LSGP model such as the vertex-frequency spectrum and the extension and the restriction. The vertex-frequency spectrum of LSGPs can be regarded as the generalization of the time-frequency spectrum concept in classical signal processing to graph domains. Apart from its theoretical properties, a learning algorithm for the membership functions and the parameters for the polynomial coefficients generating the component processes has been presented. The choice of polynomials for representing spectral kernels is common practice in graph signal processing. Then to decrease the complexity of the algorithm on large graphs, we have proposed a partitioning scheme inspired by the theoretical findings in Chapter 5. Last but not least, we have evaluated the performance of our algorithm both in synthetic and real datasets. The LSGP model gives promising results compared to other stochastic process models in graph signal processing.

LSGPs are characterized by their component processes, which determine their frequency characteristics, and by the membership functions that determine how the frequency characteristics of each model spread over the graph. If there is no restriction on the membership functions of the process, any membership function that might have abrupt changes between adjacent nodes is allowed, contradicting the locality of the process. To enforce the locality of an LSGP, the smoothness of the member-

ship functions is desired. In turn, as pointed out in Chapter 3, this creates a vertex-frequency spectrum where the frequency content changes smoothly over the whole graph. Moreover, the existence of the vertex-frequency spectrum allows the extension and the restriction of the vertex-frequency contents to other graphs having more or fewer nodes, respectively. How to sample the vertex-frequency content of an LSGP is an open question and should be investigated in future work. Future directions may include learning the LSGP in an adaptive way, i.e. updating the process when more nodes are introduced into the network or some of the nodes are deleted, or modelling stochastic processes on a large graph whose characteristics are driven by an LSGP on a smaller graph, inspired by the extension and restriction discussion.

The learning of the nonparametric model of LSGP is a complex task. Therefore, in Chapter 4, the frequency kernels are assumed to be polynomial. The learning method then consists of learning the covariance matrix for the LSGP model, by estimating the membership functions and polynomial parameters. However, this creates a polynomial of order 4 in each parameter (membership functions and polynomial parameters), making the problem non-convex. To decrease the polynomial order and make the problem in each variable a convex one, we propose a learning algorithm to learn some auxiliary variables that have a second order dependence on the membership functions and polynomial parameters. This results in two conic problems, which can be solved using primal-dual methods and Gauss-Newton method. This, however, increases the complexity of learning the parameters, which should be examined in future work. Therefore, the future work for this part involves decreasing the algorithm's complexity by introducing parametric membership functions or changing the method for estimating the parameters.

The learning algorithm used in Chapter 4 gives promising results emphasizing the necessity of vertex dependence, whose complexity is addressed by the theoretical findings in Chapter 5. Since LSGPs are inherently composed of different global models, we have analyzed the conditions under which the covariance of an LSGP is close to that of a WSS process. Consequently, we have found that the covariance of an LSGP can be easily decomposed if three conditions hold: the membership functions are concentrated on their designated subgraphs and minimal on the complements; the frequency kernels do not overlap in the frequency domain; and, finally, the kernels

exhibit lowpass characteristics. Of these three, the least significant is the lowpass behaviour of kernels, as demonstrated by the theoretical findings. Based on these findings, we have proposed a partitioning algorithm to expose subgraphs that can provide approximate supports for a single component of the LSGP. Then, from the partitioned subgraphs, LSGPs are learned accordingly. As shown in Chapter 6, although this approach is promising, the main issue is that error is accumulated on the boundaries. This problem should be explored in future research, incorporating the flow of information between partitions.

We have exhibited the effectiveness of the algorithms and the expressiveness of the model in three parts. In the first part, we have demonstrated the performance of our algorithm in a controlled environment, i.e., on a synthetic data set consisting of realizations of a Gaussian process generated with respect to our signal model. In the experiments, we conclude that the proposed learning algorithm for LSGP parameter estimation, as described in Chapter 4, is robust against the introduction of noise to the given dataset and that the effect of model complexity is predictable. A surprising discovery is that the sensitivity of the algorithm to the regularization parameters is an indicator of the smoothness of the given dataset, or, the smoothness of the associated membership functions. In the second part, we investigated the sensitivity of the partitioning algorithm in Chapter 5 to the vertex and frequency separation parameters, demonstrating that the algorithm successfully partitions the graph into subgraphs over which the data exhibits an approximately WSS behavior. In the last part, the strength of the algorithm compared to other stochastic process model methods has been demonstrated, where the LSGP algorithm is shown to have remarkable estimation performance.

In conclusion, LSGP is a novel stochastic graph process model that might pave a way for non-stationary graph signal processing with its strong properties. This makes the LSGP a promising subject to study further, where the future directions may include LSGP models for time-vertex signal processing and further investigation of its theoretical properties for gaining a deeper understanding of its behavior.

REFERENCES

[1] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, 10 2012.

[2] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Graph filters," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6163–6166, May 2013.

[3] Y. Tanaka, Y. C. Eldar, A. Ortega, and G. Cheung, "Sampling signals on graphs: From theory to applications," *IEEE Signal Processing Magazine*, vol. 37, pp. 14–30, Nov 2020.

[4] B. Girault, "Stationary graph signals using an isometric graph translation," in *23rd European Signal Processing Conference (EUSIPCO)*, pp. 1516–1520, 2015.

[5] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *IEEE Transactions on Signal Processing*, vol. 65, no. 22, pp. 5911–5926, 2017.

[6] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 65, pp. 3462–3477, July 2017.

[7] A. Loukas and N. Perraudin, "Stationary time-vertex signal processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2019, no. 1, p. 36, 2019.

[8] T. W. Anderson, *The statistical analysis of time series*. John Wiley & Sons, 2011.

[9] R. Dahlhaus, "Locally stationary processes," *Handbook of Statistics*, vol. 30, pp. 351–413, 2012.

[10] D. Angelosante and G. B. Giannakis, "Sparse graphical modeling of piecewise-stationary time series," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1960–1963, 2011.

[11] A. Hasanzadeh, X. Liu, N. Duffield, and K. R. Narayanan, "Piecewise stationary modeling of random processes over graphs with an application to traffic prediction," *Proc. IEEE International Conference on Big Data*, pp. 3779–3788, 2019.

[12] S. Sardellitti, S. Barbarossa, and P. D. Lorenzo, "On the graph fourier transform for directed graphs," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 6, pp. 796–811, 2017.

[13] F. Chung, "Laplacians and the cheeger inequality for directed graphs," *Annals of Combinatorics*, vol. 9, pp. 1–19, 2005.

[14] R. Shafipour, A. Khodabakhsh, G. Mateos, and E. Nikolova, "A directed graph fourier transform with spread frequency components," *IEEE Transactions on signal processing*, vol. 67, no. 4, pp. 946–960, 2018.

[15] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[16] L. J. Grady and J. R. Polimeni, *Discrete calculus: Applied analysis on graphs for computational science*, vol. 3. Springer, 2010.

[17] S. Shekkizhar and A. Ortega, "Graph construction from data by non-negative kernel regression," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3892–3896, May 2020.

[18] B. Girault, P. Gonçalves, and E. Fleury, "Translation on graphs: An isometric shift operator," *IEEE Signal Processing Letters*, vol. 22, pp. 2416–2420, Dec 2015.

[19] B. Scalzo, L. Stanković, M. Daković, A. G. Constantinides, and D. P. Mandic, "A class of doubly stochastic shift operators for random graph signals and their boundedness," *Neural Networks*, vol. 158, pp. 83–88, 2023.

[20] E. Isufi, A. Loukas, N. Perraudin, and G. Leus, "Forecasting time series with varma recursions on graphs," *IEEE Transactions on Signal Processing*, vol. 67, no. 18, pp. 4870–4885, 2019.

[21] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "Vertex-frequency analysis on graphs," *Applied and Computational Harmonic Analysis*, vol. 40, no. 2, pp. 260–291, 2016.

[22] W. E. Boyce, R. C. DiPrima, and D. B. Meade, *Elementary differential equations and boundary value problems*. John Wiley & Sons, 2021.

[23] F. R. Chung, *Spectral graph theory*, vol. 92. American Mathematical Soc., 1997.

[24] S. Axler, *Linear algebra done right*. Springer Science & Business Media, 1997.

[25] A. V. Oppenheim, A. S. Willsky, S. H. Nawab, and J.-J. Ding, *Signals and systems*, vol. 2. Prentice hall Upper Saddle River, NJ, 1997.

[26] J. W. Cooley, P. A. W. Lewis, and P. D. Welch, "The fast fourier transform and its applications," *IEEE Transactions on Education*, vol. 12, no. 1, pp. 27–34, 1969.

[27] K.-S. Lu and A. Ortega, "Fast graph fourier transforms based on graph symmetry and bipartition," *IEEE Transactions on Signal Processing*, vol. 67, no. 18, pp. 4855–4869, 2019.

[28] L. Le Magoarou, R. Gribonval, and N. Tremblay, "Approximate fast graph fourier transforms via multilayer sparse approximations," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 2, pp. 407–420, 2018.

[29] P. Stoica, R. L. Moses, *et al.*, *Spectral analysis of signals*, vol. 452. Pearson Prentice Hall Upper Saddle River, NJ, 2005.

[30] J. Liu, E. Isufi, and G. Leus, "Filter design for autoregressive moving average graph filters," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, pp. 47–60, March 2019.

[31] J. Mei and J. M. F. Moura, "Signal processing on graphs: Causal modeling of unstructured data," *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pp. 2077–2092, 2017.

[32] M. Nerlove, D. M. Grether, and J. L. Carvalho, *Analysis of economic time series: a synthesis*. Academic Press, 2014.

[33] R. A. Silverman, "Locally stationary random processes," *IRE Transactions on Information Theory*, vol. 3, pp. 182–187, 1957.

[34] P. Wahlberg and M. Hansson, "Kernels and multiple windows for estimation of the wigner-ville spectrum of gaussian locally stationary processes," *IEEE Transactions on Signal Processing*, vol. 55, pp. 73–84, Jan 2007.

[35] B. Boashash, *Time-frequency signal analysis and processing: a comprehensive reference*. Academic press, 2015.

[36] M. Hansson and J. Sandberg, "Multiple windows for estimation of locally stationary transients in the electroencephalogram," in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pp. 7293–7296, Jan 2005.

[37] R. Anderson and M. Sandsten, "Inference for time-varying signals using locally stationary processes," *Journal of Computational and Applied Mathematics*, vol. 347, pp. 24–35, 2 2019.

[38] S. Mallat, G. Papanicolaou, and Z. Zhang, "Adaptive covariance estimation of locally stationary processes," *The annals of Statistics*, vol. 26, no. 1, pp. 1–47, 1998.

[39] D. Donoho, S. Mallat, and R. von Sachs, "Estimating covariances of locally stationary processes: consistency of best basis methods," in *Proceedings of Third International Symposium on Time-Frequency and Time-Scale Analysis (TFTS-96)*, pp. 337–340, June 1996.

[40] F. Roueff and R. V. Sachs, "Time-frequency analysis of locally stationary hawkes processes," *https://doi.org/10.3150/18-BEJ1023*, vol. 25, pp. 1355–1385, 5 2019.

[41] R. Dahlhaus, "Fitting time series models to nonstationary processes," *https://doi.org/10.1214/aos/1034276620*, vol. 25, pp. 1–37, 2 1997.

[42] R. Anderson and M. Sandsten, "Classification of eeg signals based on mean-square error optimal time-frequency features," in *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 106–110, Sep. 2018.

[43] J. Pitton, "Adapting multitaper spectrograms to local frequency modulation," in *Proceedings of the Tenth IEEE Workshop on Statistical Signal and Array Processing (Cat. No.00TH8496)*, pp. 108–112, Aug 2000.

[44] W. Palma and R. Olea, "An efficient estimator for locally stationary gaussian long-memory processes," *The Annals of Statistics*, vol. 38, no. 5, pp. 2958–2997, 2010.

[45] F. Hua, R. Nassif, C. Richard, H. Wang, and A. H. Sayed, "Decentralized clustering for node-variant graph filtering with graph diffusion lms," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pp. 1418–1422, Oct 2018.

[46] F. Gama, G. Leus, A. Marques, and A. Ribeiro, "Convolutional neural networks via node-varying graph filters," pp. 1–5, 06 2018.

[47] S. Segarra, A. G. Marques, and A. Ribeiro, "Distributed implementation of linear network operators using graph filters," *Allerton Conference on Communication, Control, and Computing*, pp. 1406–1413, 2016.

[48] S. Segarra, A. G. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 4117–4131, 2017.

[49] F. Gama, B. G. Anderson, and S. Sojoudi, "Node-variant graph filters in graph neural networks," in *2022 IEEE Data Science and Learning Workshop (DSLW)*, pp. 1–6, IEEE, 2022.

[50] L. Stankovic, "An analysis of some time-frequency and time-scale distributions," in *Annales des telecommunications*, vol. 49, pp. 505–517, Paris, Societe de la Revue optique., 1994.

[51] K. Gröchenig, *Foundations of time-frequency analysis*. Springer Science & Business Media, 2001.

[52] E. Sejdić, I. Djurović, and J. Jiang, "Time–frequency feature representation using energy concentration: An overview of recent advances," *Digital signal processing*, vol. 19, no. 1, pp. 153–183, 2009.

[53] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[54] B. Girault, S. S. Narayanan, and A. Ortega, "Towards a definition of local stationarity for graph signals," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4139–4143, 2017.

[55] D. Thanou, X. Dong, D. Kressner, and P. Frossard, "Learning heat diffusion graphs," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 3, pp. 484–499, 2017.

[56] L. Stanković, D. Mandic, M. Daković, B. Scalzo, M. Brajović, E. Sejdić, and A. G. Constantinides, "Vertex-frequency graph signal processing: A comprehensive review," *Digital Signal Processing*, vol. 107, p. 102802, 2020.

[57] B. Girault, S. S. Narayanan, and A. Ortega, "Local stationarity of graph signals: insights and experiments," *SPIE*, p. 60, 2017.

[58] G. P. H. Styan, "Hadamard products and multivariate statistical analysis," *Linear Algebra and its Applications*, vol. 6, pp. 217–240, 1973.

[59] K. Toh, R. H. Tütüncü, and M. Todd, "Inexact primal-dual path-following algorithms for a special class of convex quadratic sdp and related problems," *Pacific Journal of Optimization*, vol. 3, 04 2006.

[60] K.-C. Toh, M. Todd, and R. Z, "Sdpt3—a matlab software package for semidefinite programming, version 2.1," *Optimization Methods & Software - OPTIM METHOD SOFTW*, vol. 11, 10 1999.

[61] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control* (V. Blondel, S. Boyd, and

H. Kimura, eds.), Lecture Notes in Control and Information Sciences, pp. 95–110, Springer-Verlag Limited, 2008.

[62] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1." `http://cvxr.com/cvx`, Mar. 2014.

[63] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[64] X. Li, D. Sun, and K.-C. Toh, "Qsdpnal: A two-phase augmented lagrangian method for convex quadratic semidefinite programming," *Mathematical Programming Computation*, vol. 10, pp. 703–743, 2018.

[65] K.-C. Toh, "An inexact primal–dual path following algorithm for convex quadratic sdp," *Mathematical Programming*, vol. 112, pp. 221–254, Mar 2008.

[66] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.

[67] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, "On the implementation and usage of sdpt3–a matlab software package for semidefinite-quadratic-linear programming, version 4.0," *Handbook on semidefinite, conic and polynomial optimization*, pp. 715–754, 2012.

[68] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *Journal of Control and Decision*, vol. 5, no. 1, pp. 42–60, 2018.

[69] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM review*, vol. 38, no. 1, pp. 49–95, 1996.

[70] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[71] F. Alizadeh, "Interior point methods in semidefinite programming with applications to combinatorial optimization," *SIAM journal on Optimization*, vol. 5, no. 1, pp. 13–51, 1995.

[72] R. M. Damerell, "On moore graphs," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 74, no. 2, p. 227–236, 1973.

[73] C. C. Ni, Y. Y. Lin, F. Luo, and J. Gao, "Community detection on networks with ricci flow," *CoRR*, vol. abs/1907.03993, 2019.

[74] E. T. Guneyi, B. Yaldiz, A. Canbolat, and E. Vural, "Learning graph arma processes from time-vertex spectra," *arXiv preprint arXiv:2302.06887*, 2023.

[75] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, p. 066111, 2004.

[76] "Covid-19 coronavirus pandemic data."

[77] "Eurostat: An official website of the european union."

[78] A. Arguez, I. Durre, S. Applequist, R. S. Vose, M. F. Squires, X. Yin, R. R. Heim, and T. W. Owen, "Noaa's 1981–2010 u.s. climate normals: An overview," *Bulletin of the American Meteorological Society*, vol. 93, no. 11, pp. 1687 – 1697, 2012.

[79] A. Jung, A. O. Hero, III, A. C. Mara, S. Jahromi, A. Heimowitz, and Y. C. Eldar, "Semi-supervised learning in network-structured data via total variation minimization," *IEEE Transactions on Signal Processing*, vol. 67, no. 24, pp. 6256–6269, 2019.

[80] M. Nagahama, K. Yamada, Y. Tanaka, S. H. Chan, and Y. C. Eldar, "Graph signal restoration using nested deep algorithm unrolling," *IEEE Transactions on Signal Processing*, vol. 70, pp. 3296–3311, 2022.

[81] H. E. Beck, N. E. Zimmermann, T. R. McVicar, N. Vergopolan, A. Berg, and E. F. Wood, "Present and future köppen-geiger climate classification maps at 1-km resolution," *Scientific data*, vol. 5, no. 1, pp. 1–12, 2018.

[82] J. Franklin, F. W. Davis, M. Ikegami, A. D. Syphard, L. E. Flint, A. L. Flint, and L. Hannah, "Modeling plant species distributions under future climates: how fine scale do climate projections need to be?," *Global change biology*, vol. 19, no. 2, pp. 473–483, 2013.

[83] R. P. Tollo, L. Corriveau, J. McLelland, and M. J. Bartholomew, “Proterozoic tectonic evolution of the grenville orogen in north america: An introduction,” *Geological Society of America Memoirs*, vol. 197, pp. 1–18, 2004.