The geometry of signal and image patch-sets

by

K. M. Taylor

B.S./M.S., University of Colorado at Boulder, 2008

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Applied Mathematics

2011

This thesis entitled:
The geometry of signal and image patch-sets
written by K. M. Taylor
has been approved for the Department of Applied Mathematics

Prof. François G. Meyer

Prof. James H. Curry

Prof. Juan G. Restrepo

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Taylor, K. M. (Ph.D., Applied Mathematics)

The geometry of signal and image patch-sets

Thesis directed by Prof. François G. Meyer

In this thesis, we study the representation of local, or fine scale, snippets — or *patches* — that are extracted from a signal or image. We describe a method that characterizes the dimensionality that is observed in the set of patches when they are regarded as points in Euclidean space. Our approach is based on the assumption that the signal or image is composed of solutions to ordinary differential equations of a certain class.

We also provide a theoretical interpretation — via graph models — that explains the success of analyzing signal and image patches using diffusion-based graph metrics. Our framework is built on the assumption that there exists a partition of the signal or image's patches. Specifically, we assume there are two subsets of patches. One set comprises patches that are connected through some type of coherence in the domain of the signal, such as temporal coherence in time series, or spatial coherence between patches in the image plane. The other set comprises patches whose edge connections are not so largely influenced by the aforementioned coherence. Instead, these connections are more sporadic, with little relationship between the locations in the signal or image domain from which the patches were extracted. Using the commute time metric — a diffusion-based graph metric — we prove that the average proximity between patches in the first set grows faster than the average proximity between patches in the second set, as the number of patches approaches infinity. Consequently, a parametrization of the patches based on commute times will relatively cluster the second set of patches, which is the first step toward solving a larger problem, such as classification or clustering of the patches, detection of anomalies, or segmentation of an image.

In addition to our theoretical results, this thesis also evaluates numerical procedures designed to efficiently compute the spectral decomposition of large matrices. These procedures include the

Nyström extension [24], and a multilevel eigensolver. Finally, we benchmark a classifier that is trained on the commute time embedding of a dataset of seismic events, against a standard algorithm used to detect arrival-times.

Dedication

To my family.

Acknowledgements

I am extraordinarily fortunate to have worked with and learned from so many wonderful people. I would not be where I am if it were not for my advisor, Prof. Meyer. I cannot tell you thanks enough for all your guidance and support. In addition, Prof. Curry has given me more than I deserve; he has been like a rock for me. Also, I cannot thank Prof. Lladser, Prof. Preston, and Prof. Restrepo enough for encouraging me and for inspiring me personally and academically. I am lucky to learn from teachers who respect the subject so much. Finally, I must thank Prof. Dougherty for getting me excited enough to begin this whole math adventure, and for always being there when I needed her.

To everyone in the applied math department, faculty and staff, I am so grateful for the ways in which you have touched my life.

Finally, if not for the strength and support of my family and friends, I would be nothing. Thank you for everything.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

Obtaining a useful representation of a dataset is fundamental in many applications. The goal is that the organization of the data in the new representation will make its meaningful characteristics or properties easier to recognize for purposes such as learning about the system which generated the data, detecting features of interest, prediction, compression, or classification.

1.1 Using graphs to represent datasets

Graphs, or networks, provide a convenient way to represent a dataset because they are simple to construct, yet capable of encoding complex interactions, similarities, or patterns which exist in the data. One standard way to map a dataset to a graph is to identify vertices of the graph as data points, while (weighted) edges of the graph identify similarities between the data points. The *geometry* of the graph is the set of relationships between its vertices, including the relationship of being connected with an edge. The geometry of a dataset's graph representation has been used to study the spread of disease, the internet, polymers, etc.

In addition to the obvious vertex interaction that is indicated by an edge, more complex or subtle interactions can be discovered by studying other properties of the graph's geometry. Such properties include clustering coefficients, vertex centrality, and degree distributions. Other properties of the graph's geometry can also be inferred from the set of eigenvalues corresponding to matrices associated with the graph, including the adjacency matrix, the graph's discrete Laplace operator (a.k.a the graph Laplacian), or the normalized graph Laplacian [20].

Diffusion-based graph metrics. In contrast to static attributes of the graph geometry, dynamical processes are also used to reveal relationships between the graph's vertices. Specifically, one can define a diffusion process or a random walk on the graph (see section 4.2) that provides an intrinsic multiscale approach to understanding the graph's geometry; as time evolves, starting from an initial vertex of the graph, the process is more likely to "visit" a larger subset of the graph's vertices. This notion is quantified by defining metrics on the graph that are based on the diffusion process. These metrics can be used to compare vertices of the graph and add to our understanding of the graph's geometry.

The normalized graph Laplacian mentioned before also plays a role in defining the diffusion equation on the graph, analogous to the Laplace operator in Euclidean space. Indeed, the spectral decomposition of the normalized graph Laplacian is used to expand solutions to diffusion equations defined on the graph. Thus, analogous to classical Fourier analysis, the eigenfunctions of the normalized graph Laplacian can be related to a notion of smoothness that is adapted to the geometry of the graph [81]. Moreover, the spectral decomposition of the normalized graph Laplacian can be used to express a variety of diffusion-based distances between vertices (see section 4.2).

Graph parametrizations. In order to replace the abstract geometry of the graph with a more concrete geometry where subsequent analysis of the dataset can be performed, vertices of the graph are mapped to points in Euclidean space. In this space, the distance between vertices encodes the diffusion-based metric. Embedding a finite metric space in a Euclidean space is nontrivial, and in general, not possible. [62]. However, the diffusion-based metrics we consider in this work can be preserved exactly through an embedding into Euclidean space provided the graph is connected and the Markov process defined in chapter 4 is aperiodic.

Techniques used to map the graph representation of the dataset into a new Euclidean space are similar to techniques used to embed manifolds. Informally, a manifold is a generalization of the concept of a surface in three dimensions. Analogous to a linear space, a manifold has a dimension that characterizes the number of distinct coordinates that are needed to fully parametrize the manifold. For instance, the sphere in three dimensions is a two dimensional manifold because every

point on the sphere can be described uniquely by its latitude and longitude. In addition, every small part of a j-dimensional manifold can be smoothly mapped to $\mathbb{R}^j$. Examples of manifolds include a sphere, a torus, a cone, and the set of all k-dimensional subspaces of $\mathbb{R}^l$, with $k \leq l$.

The relationship between embedding manifolds and graphs is utilized in highly successful dataset parametrizations, including ISOMAP [86], local linear embedding [73], Laplacian eigenmaps [7], and diffusion maps [23]. The number of studies which demonstrate the ability of the aforementioned algorithms to recover complex interactions between data points is growing, in part due to the empirical success of such studies. Adding to the popularity and success of algorithms such as ISOMPAP, LLE, Laplacian eignmaps and diffusion maps, the objects that are generated by the algorithms are guaranteed to converge to invariant objects on the underlying manifold [23].

Signal and image data Despite the observed success of the dataset embedding techniques whose convergence guarantees rely on a smooth manifold generating the data [73, 7, 23, 86], for many data types, it is unclear if there is truly an underlying manifold which satisfies such assumptions. Signals and images are two such data types. In fact, Wakin et al. describe a representation of an image model that does not satisfy the smooth manifold assumption in [93]. Nevertheless, works that parametrize signals or images and use procedures which rely on the manifold hypothesis to prove convergence are both prevalent and effective [56, 92, 16, 81, 80]. In fact, small images are parametrized in each of the papers [73, 23, 86].

The effectiveness of the parametrizations [73, 7, 23, 86] that are meant to recover low-dimensional structure in signal and image data is intuitive. That is, if the image is of a natural scene or if the signal is the recording of a human voice, for example, then there is some underlying set of constraints that limit the possible configuration of content in the signal or image. The captured scene limits the distribution of an image's pixels, while physical constraints limit the sounds a human can produce. Consequently, signal and image data can be represented using very few parameters because the set of patches has very few "degrees of freedom."

Formalizing this intuition is the motivation behind the development of concise signal and image models. Such models are typically defined on, or emphasize, the local (or fine) scale of the

signal or image. This is a natural framework; although two images of natural scenes may have no correlation whatsoever, the frequent occurrence of edges, repeated patterns, and large smooth regions in natural scenes leads to local correlations between two different images. The same logic applies to signals.

1.2 Contribution and structure of this thesis

In this thesis, we study the graph representation of local, or fine scale, blocks, or snippets, that are extracted from inside a signal or image. When processing signals, a vertex can be thought of as a sliding temporal window, for instance. Similarly, when processing images, a vertex can be thought of a 8-by-8 pixel block, for example. We refer to these snippets as *patches*.

First, we describe a method that characterizes the dimensionality, or "degrees of freedom," that is observed in the set of patches when the patches are regarded as points in Euclidean space — before being mapped to a graph. Our approach is based on the assumption that the signal or image is composed of solutions to linear constant-coefficient, homogeneous, ordinary differential equations. Solutions of this kind include decaying oscillations, a prevalent feature in many signals and images. Furthermore, we present corollaries indicating that our results can be used locally and approximately, in the sense that if only a portion of a signal or image can be approximated by an ODE's solution, then we can characterize the dimensionality of the point cloud of patches associated with that portion of the signal or image.

Second, we provide a theoretical interpretation — via graph models — that explains the success of diffusion-based graph embeddings of signal and image patches. Although we motivate our study using the graph representation of signal and image patches in particular, our theoretical conclusions on the graph's embedding are completely general. Our framework is built on the assumption that there exists a partition of the signal or image's patches. In particular, we assume there are two subsets of patches. One set comprises patches that are connected through some type of coherence in the domain of the signal, such as temporal coherence in time series, or spatial coherence between patches in the image plane. The other set comprises patches whose edge connections are

not so largely influenced by the aforementioned coherence. Instead, these connections are more sporadic, with little relationship between the locations in the signal or image domain from which the patches were extracted. Using the commute-time metric — a diffusion-based graph metric — we prove that the average proximity between patches in the first set grows faster than the average proximity between patches in the second set, as the number patches approaches infinity. Consequently, a parametrization of the patches based on commute-times will relatively cluster the second set of patches, which is the first step toward solving a larger problem, such as classification or clustering of the patches, detection of anomalies, or segmentation of an image.

In addition to our theoretical results, this thesis also evaluates numerical procedures designed to efficiently compute the spectral decomposition of large matrices. These procedures include out-of-sample extension via the Nyström extension [24], and a multilevel technique based on algebraic multigrid [e.g. 88, and references therein]. Both of these techniques require choosing a subset of the patches from which to interpolate. We find that choosing this subset is critical to performance, even on toy datasets. Finally, we demonstrate that a commute-time parametrization is able to organize very large, real datasets that have a high degree of variability. Specifically, in the attempt to automatically identify arrival times of seismic waves, we benchmark a classifier that is trained on the commute-time embedding of a dataset of seismic events, against an optimized *picker*, which is a standard algorithm used to detect arrival-times of incoming seismic waves. The classifier trained on the commute-times outperforms this optimized picker in detecting arrivals in unseen seismic traces, as well as the same type of classifier trained on a principal component analysis and a wavelet analysis of the seismic dataset.

This thesis is organized as follows (see Figure 1.1). In chapter 2, we briefly review related works, while establishing preliminaries, including notation and definitions. In chapter 3, we present our theoretical conclusions on the organization of a signal or image's set of patches as points in Euclidean space. In chapter 4, we present our graph models and theoretical results and analysis of the commute-times on these graph models. We show that the commute-time embedding is able to efficiently organize the seismic dataset for the purpose of arrival-time estimation in chapter 5. Our

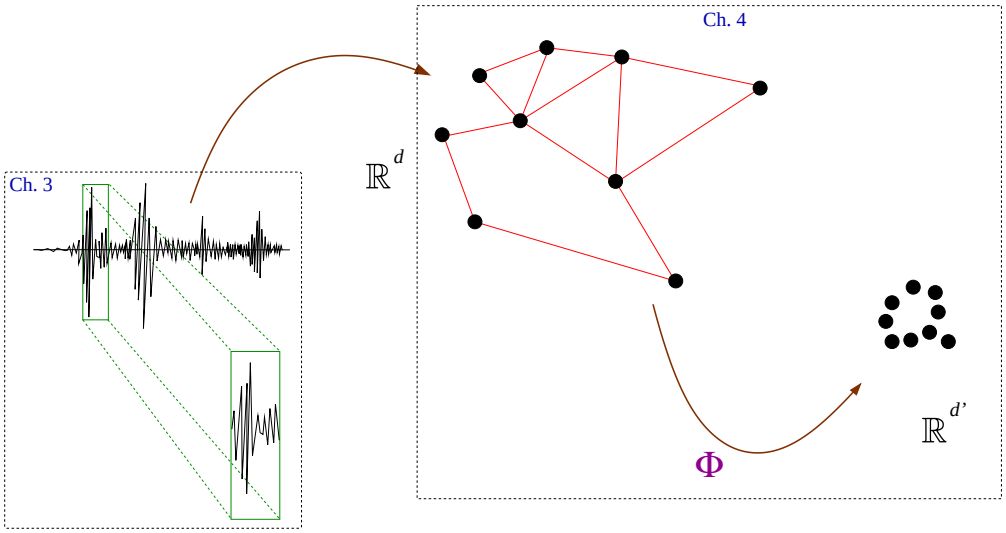

Figure 1.1: The main contribution of this thesis is in chapters 3 and 4. Chapter 3 studies the organization of a signal or image's patch-set as points in $\mathbb{R}^d$, while chapter 4 focuses on the patch-graph and its embedding into $\mathbb{R}^{d'}$.

evaluation of numerical methods for rapidly computing the normalized graph Laplacian's spectral decomposition is given in chapter 6. We conclude, and provide both a general interpretation of our results, and questions triggered by this work in chapter 7.

Chapter 2

A review of existing work

2.1 Introduction

Analyzing a dataset via diffusion on its graph representation is a highly successful approach, especially if that dataset comprises local properties of signals and images. In this section, we describe several related works that utilize, model, or statistically study the geometry associated with a graph of *patches* — or local snippets — of a signal or image. We also describe related works that only consider the patches as points in Euclidean space. Such works offer critical insight since the edges of the graph are typically constructed based on proximity of the patches as points in space.

2.2 Local analysis of signals and images using patches

We begin with several definitions. For simplicity and without loss of generality, we define concepts using only a signal that is formed by a time series, or a sequence of samples, $\{x_n\}_{n=1}^{N'}$. Because we want to extract $N = N' - (d-1)$ patches from this sequence, we need d extra samples at the end (hence the N' samples). We first define the notion of a *patch*.

Definition 1. *We define a* ***patch*** *as a vector in* $\mathbb{R}^d$ *formed by a subsequence of* $d \geq 1$ *contiguous samples extracted from the time series,*

$$\mathbf{x}_n = (x_n, x_{n+1}, \ldots, x_{n+d-1}), \quad \text{for } n = 1, 2, \ldots, N, \tag{2.1}$$

See Figure 2.1 for an illustration.

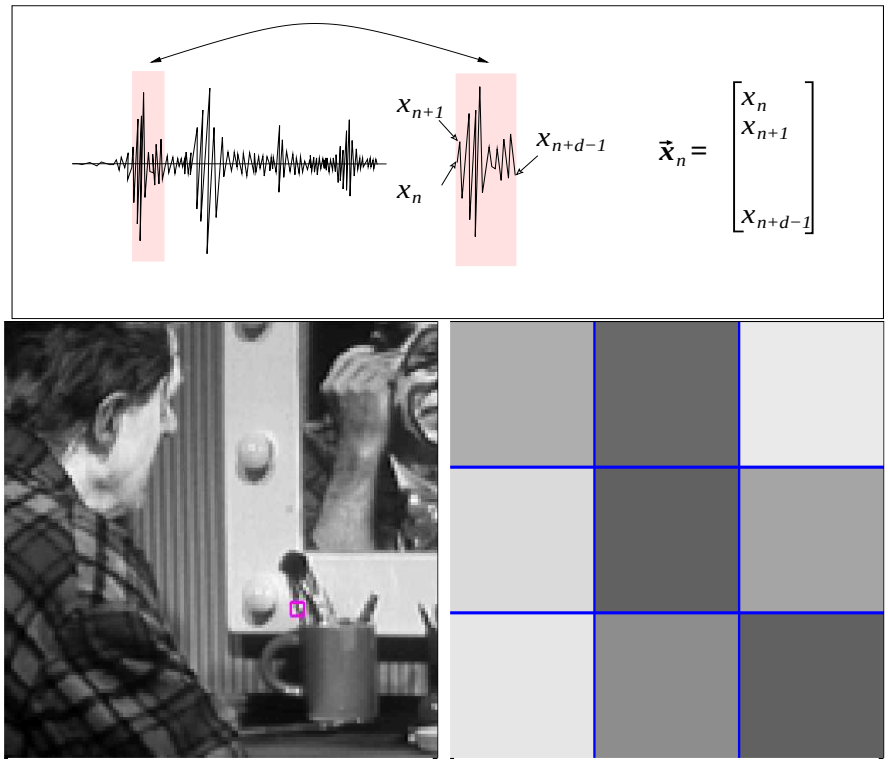

Figure 2.1: Top: A patch $\mathbf{x}_n$ extracted from a time series $\{x_n\}$ is composed of d equally-spaced time samples. Bottom: A patch extracted from an image is a square block of pixel intensities. For example, an outline of a 3-by-3 patch in pink (on the left), and a closer view of the same patch (on the right).

As we collect all the patches, we form the *patch-set* in $\mathbb{R}^d$.

Definition 2. *The **patch-set** is defined as the set of patches extracted from the time series,*

$$\textit{patch-set} = \{\mathbf{x}_n, n = 1, 2, \ldots, N\}. \tag{2.2}$$

In order to study the discrete structure formed by the patch-set (2.2), we consider the patch-graph Γ defined as follows.

Definition 3. *We define the **patch-graph**, Γ, as a weighted graph such that:*

(1) The set of vertices is formed by the patch-set $\{\mathbf{x}_n, n = 1, \ldots, N\}$.

(2) Each vertex $\mathbf{x}_n$ is connected to its ν nearest neighbors using a fixed metric, ρ, that is defined on $\mathbb{R}^d$. If there are multiple neighbors which are equidistance apart, then we randomly order

these neighbors as first, second, etc. Using this ordering when necessary, we can associate ν neighbors with each vertex $\mathbf{x}_n$.

(3) The weight $w_{n,m}$ along the edge $\{\mathbf{x}_n, \mathbf{x}_m\}$ is given by

$$w_{n,m} = \begin{cases} e^{-\rho^2(\mathbf{x}_n, \mathbf{x}_m)/\sigma^2} & \text{if } \mathbf{x}_n \text{ is connected to } \mathbf{x}_m \text{ or if } \mathbf{x}_m \text{ is connected to } \mathbf{x}_n, \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

The edges of the patch-graph encode the similarities between its N vertices: a large weight $w_{n,m}$ occurs only if $\mathbf{x}_n$ and $\mathbf{x}_m$ are close with respect to ρ. The parameter σ controls the influence of the similarity on the edge weight $w_{n,m}$. In particular, $w_{n,m}$ will be significant only when $\rho(\mathbf{x}_n, \mathbf{x}_m)$ is comparable in magnitude to, or much smaller than, σ.

We observe that the nearest neighbor relationship in $\mathbb{R}^d$ is not necessarily symmetric, however after establishing edges, the definition of the weights in (2.3) leads to an undirected graph. That is the edge $\{\mathbf{x}_n, \mathbf{x}_m\}$ is also identified with the edge $\{\mathbf{x}_m, \mathbf{x}_n\}$. We relate the choice of metric ρ to the normalization of the patches, as described in section 3.5.

Although we define the patch-graph as a ν nearest neighbor graph, there are alternative ways to construct a graph of patches. For example, a graph could be defined that considers only edges connecting patches that are within some radius ϵ.

We use a ν nearest neighbor graph because a key component of our theoretical result of chapter 4 relies on recovering the local structure in the patch set. To appreciate such structure in a ϵ ball graph, the neighborhood radius ϵ must be chosen small enough not to blur the local structure into one large component of the graph. However, such a small neighborhood could result in a disconnected graph since patches who are farther than ϵ away from every other patch would not be connected to other patches. We must avoid a disconnected graph because it is contrary to the assumptions of the analysis in the following sections. Although we cannot guarantee that a ν nearest neighbor graph is connected, we have observed in practice that a ν nearest neighbor graph is less likely to be disconnected than an ϵ ball graph. See Figure 2.2 for an illustration.

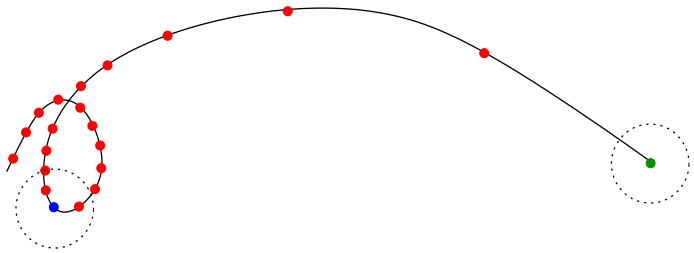

Figure 2.2: An illustration of a patch-set with different local structures at different times and the consequence of using an ϵ-neighborhood graph. Discrete patches are represented by red dots. Two ϵ-neighborhoods of points are indicated by dotted lines. Because we want to recognize the low-dimensional structure in regions where patches are close together, we would require ϵ be small enough (see blue point). However, this leads to disconnected components of the graph (see green point).

Lastly, observe that the weighted graph is fully characterized by its *weight matrix*.

Definition 4. *The* ***weight matrix*** $\mathbf{W}$ *is the* $N \times N$ *matrix with entries* $\mathbf{W}_{n,m} = w_{n,m}$. *The* ***degree matrix*** *is the* $N \times N$ *diagonal matrix* $\mathbf{D}$ *with entries* $\mathbf{D}_{n,n} = \sum_{l=1}^{N} w_{n,l}$.

2.2.1 Dynamical systems analysis

The concept of patches is equivalent to the concept of *time-delay coordinates* in the context of dynamical systems [75, 1, 43]. More precisely, when a dynamical system – i.e. a system of ordinary differential equations (or partial differential equations) – describes a physical process and gives rise to the time series $\{x_n\}$, then Taken's embedding theorem [82] allows us to replace the unknown *phase space* of the dynamical system with a topologically equivalent phase space formed by the patch-set $\{\mathbf{x}_n\} \subset \mathbb{R}^d$. In plain English, we can learn about a complicated dynamical system by simply observing the evolution of a vector of d consecutive measurements (as in (2.1)) from the dynamical system. We note that there exists a rich literature on the analysis of geophysical time series using this concept of time-delay coordinates [e.g., 19, 44, 34, 35, 49, 50, 87, 27, 96, 26].

Recurrence quantification analysis (RQA) is a method to analyze the substitute phase space formed by a dynamical system's patch-set. The method constructs a *recurrence plot* in the form of

a N-by$N$ square matrix $\mathbf{R}$ with entries

$$(\mathbf{R})_{n,m} = \begin{cases} 1 & \text{if } \|\mathbf{x}_n - \mathbf{x}_m\| \leq \epsilon, \\ 0 & \text{otherwise,} \end{cases}$$

where $\|\cdot\|$ is the Euclidean norm on $\mathbb{R}^d$. The matrix $\mathbf{R}$ indicates when a dynamical system exhibits recurrent behavior and serves as a figurative fingerprint for the underlying dynamical process [31, 61].

The different patterns in the nonzero entries in $\mathbf{R}$, referred to as its *typologies* in [31], can be linked to specific characteristics of the process that generated the data, such as its periodicity, stationarity, and chaosticity. These typologies include the lengths of vertical or diagonal lines and the number of nonzero entries in $\mathbf{R}$. Indeed, the number of nonzero entries is related to the Grassberger-Procaccia correlation sum, which is used to estimate the correlation dimension of the dynamical system's attractor [46].

Using the recurrence plot generated by RQA, a relatively recent and straightforward way to recover the structure in phase space is to regard the nonzero entries of the matrix $\mathbf{R}$ as the adjacency matrix of a graph, similar to the work [40]. Note that this graph construction is similar to the construction of the patch graph Γ, except that this graph has edges that are truncated based on an ϵ radius. Other alternatives for mapping a time series to a graph include partitioning phase space into regions and then identifying these regions as vertices [13], or using other metrics for determining edge connections between vertices of the graph [98, 79, 60]. Regardless, such work has indicated that properties of the resulting graph, including the average path length, degree distribution, vertex centrality, and clustering coefficient, can be used to distinguishing nontrivial features and characteristics in the time series. In fact, similar predictive power has been demonstrated on graphs that are not related to a dynamical system's phase space in any obvious way [53].

2.2.2 Natural image statistics

Several studies on the statistics of patches extracted from natural images [66, 69, 54] have demonstrated that image patch data is highly non-Gaussian and tends to cluster around nonlinear

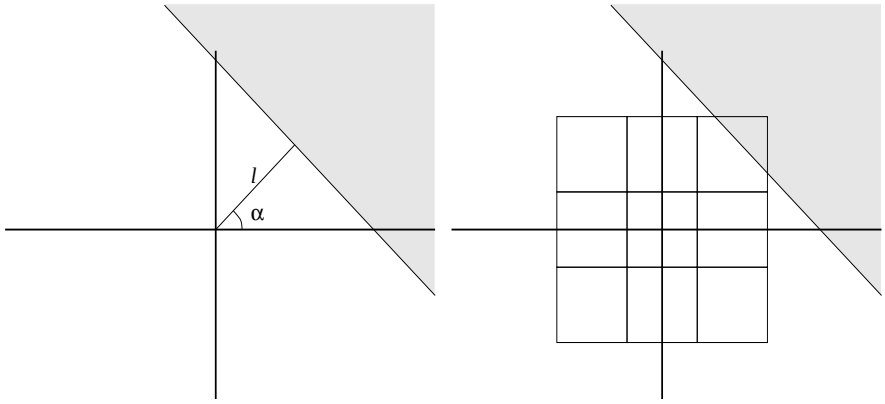

Figure 2.3: An illustration of the ideal patch manifold that is similar to the one considered in [54]. Left: The parameters l and α describe the orientation of the step edge. Right: To construct a 3-by-3 ideal patch, one averages over the underlying scene inside each square. Averaging simulates the process of discretizing a scene with pixels.

manifolds in the Euclidean space. For example, in [54], a manifold model meant to account for the nonlinear structure in the patch-set is constructed. In this model, an image patch in the set of all image patches that contain step-edges at various orientations is parametrized by the smallest angle, α, between the step-edge and the patch's relative horizontal axis, and the distance, l, between the step-edge and the patch's origin. This results in a two-dimensional manifold, with α and l as the intrinsic coordinates (see Figures 2.3 and 2.4 for more details).

As demonstrated in [54], image patches extracted from natural images lie close to the patch-set generated by this manifold model, which, considering normalization of the patches, constitutes a nonlinear two-dimensional surface embedded on the 7-dimensional sphere in $\mathbb{R}^9$.

2.2.3 Local PCA of the patch-set

As described in chapter 1, developing concise, or efficient representations for signals and images is a growing trend. These models include *sparse* representations, in which a dataset is represented using at most a fixed number of atoms or elements from a dictionary or basis [58]. An emerging idea in developing sparse representations is to construct bases or dictionaries that are adapted to the types of signals and images being processed. The K-SVD algorithm is one of these algorithms that adaptively constructs a dictionary that is able to efficiently represents a set

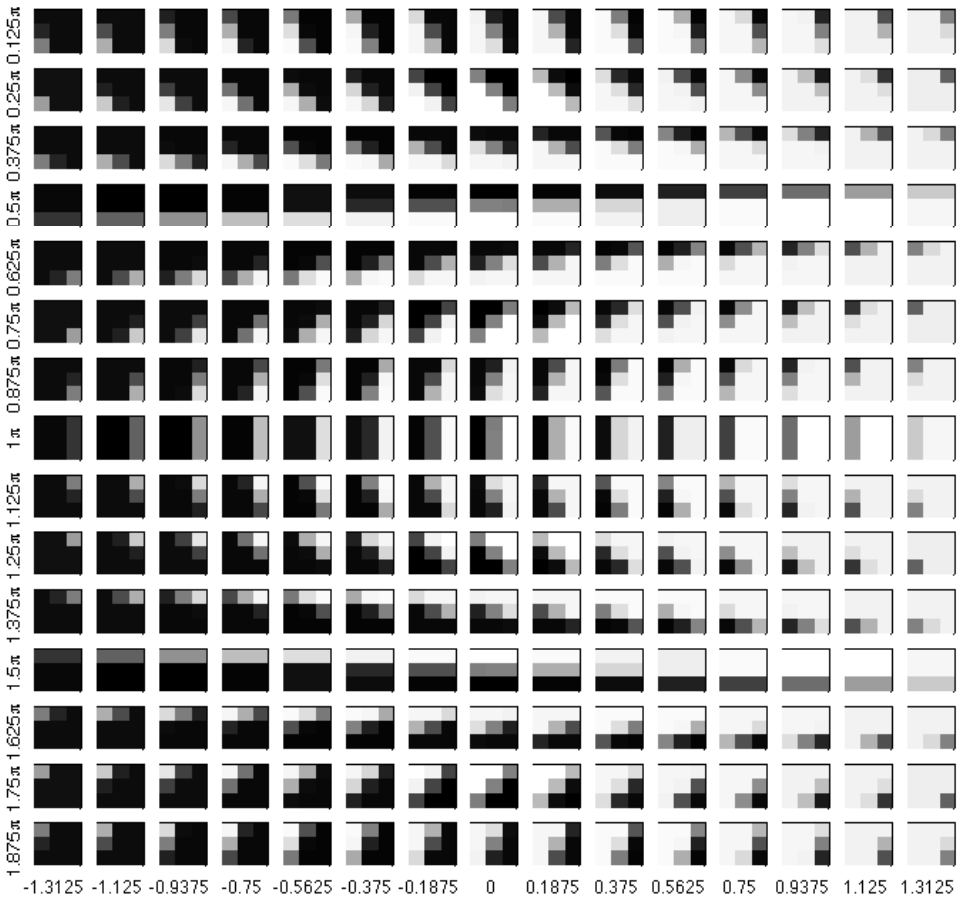

Figure 2.4: A sampling of 3-by-3 patches containing ideal step edges. The parameter l changes in the horizontal direction, while α changes in the vertical direction.

of image patches [3]. Formally, let $\mathbf{C}$ represent a d-by-K matrix, where each column represents an atom in the dictionary. In this way, we can identify $\mathbf{x}_n$ with a vector $\mathbf{y}_n \in \mathbb{R}^K$ such that

$$\mathbf{x}_n \approx \mathbf{C}\mathbf{y}_n. \tag{2.4}$$

The K-SVD algorithm searches for columns of $\mathbf{C}$ and vectors $\mathbf{y}_n$ to minimize the Euclidean norm of the error in representing the set of patches $\{\mathbf{x}_n\}_{n=1}^N$ with the set $\{\mathbf{y}_n\}_{n=1}^N$, subject to the constraint that the number of nonzero elements in each $\mathbf{y}_n$ is at most a fixed constant. This constraint ensures that the patch-set is *sparse* in the dictionary represented by $\mathbf{C}$.

The K-SVD algorithm alternates between modifying the set $\{\mathbf{y}_n\}$ and columns of $\mathbf{C}$. To compute $\mathbf{C}$, the algorithm essentially performs local principal component analysis (PCA) on the neighborhood of each patch $\mathbf{x}_n$ (see section 3.7.3 for more on local PCA). Then, columns of $\mathbf{C}$ are selected as a minimal set of components from each patch's local PCA [3]. In effect, the K-SVD algorithm amounts to *quantizing*, or *encoding*, the patch-set with columns of the dictionary $\mathbf{C}$. Note that the columns of $\mathbf{C}$ can also be regarded as patches in $\mathbb{R}^d$, and the approximation (2.4) writes every patch in the patch-set as some linear combination of the patches defining the columns of $\mathbf{C}$. From this perspective, the columns of $\mathbf{C}$ can be regarded as an ideal subset of patches from which to construct any patch in the patch-set.

2.3 Diffusion on the patch-graph

In this section, we focus specifically on related works that exploit the diffusion process on the patch-graph.

The authors [16] describe a simple method for removing noise from images. Their approach finds patches in the image that are similar to $\mathbf{x}_n$ (as measured by the metric in $\mathbb{R}^d$, for example), then obtains an estimate for a noise-free $\mathbf{x}_n$ as a weighted average of those similar patches and $\mathbf{x}_n$ itself — a type of collaborative filtering.

In [81], Szlam et al. reinterpret the nonlocal means algorithm of [16] as diffusion on the patch-graph. Specifically, a random walk is defined on the graph as follows. A random walker at vertex $\mathbf{x}_n$ will transition to a neighboring vertex $\mathbf{x}_m$ with probability

$$\mathbf{P}_{n,m} = \frac{w_{n,m}}{\sum_l w_{n,l}} = \frac{\mathbf{W}_{n,m}}{\mathbf{D}_{n,n}}. \tag{2.5}$$

We see that the random walker is very likely to transition from one patch to another that is very close, since close proximity leads to a large edge weight $w_{n,m}$. If a $\sqrt{N}$-by-$\sqrt{N}$ pixel image is represented as an N-dimensional column vector, and the matrix $\mathbf{P}$ has entries that are given in (2.5), then multiplying the column vector on the left by $\mathbf{P}$ is equivalent to evolving the diffusion process on the patch-graph for a small time step. Evolving the diffusion process on the patch-

graph removes high frequency oscillations in functions defined on the patch-graph, namely pixel intensities associated with each vertex. This process removes noise in a way analogous to classical noise removal methods.

In addition to state-of-the-art noise removal techniques, the patch-graph can also be used to segment images. For example, in [78], Shi and Malik use a graph whose vertices are functions of the patches $\mathbf{x}_n$. This graph is segmented based on values of the smallest eigenvectors associated with the smallest nonzero eigenvalue of the graph Laplacian. Despite this connection, Shi and Malik justify their approach since it minimizes the normalized cut criterion; a partition of vertices that minimizes this criterion is theoretically the most efficient in terms of minimizing the number of edges removed relative the size of the partitions created, as shown in [78].

In addition, diffusion on the patch-graph has been used in texture analysis/synthesis [56], multi-modal image registration [92], and super-resolution [71].

2.4 Conclusion

The works in the previous sections demonstrate that the patch-sets associated with signals and images exhibit low-dimensional, nonlinear geometry when regarded as points in Euclidean space. Furthermore, numerous works demonstrate that the geometry of the patch-graphs associated with signals and images can be used to improved methods for compression, classification, noise removal, and discovering the underlying dynamics behind a physical process.

A main purpose of this thesis is to add to our theoretical understanding of *(i)* the structure of the patch-set in Euclidean space and *(ii)* the intrinsic geometry of the patch-set when parametrized using embeddings of the patch-graph. We consider embeddings of the graph that typically assume that the image and signal patch data lies on some smooth manifold. However, we provide theoretical understanding of these embeddings' behaviors *without* assuming the existence any underlying smooth manifold.

Chapter 3

Extrinsic organization of signal and image patches

3.1 Introduction

A signal's patch-set can be identified as the discretization of a vector valued function that depends on time. Similarly, an image's patch-set is the discretization of another vector valued function that depends on two coordinates which span the image plane. In this chapter, we identify these vector valued functions and characterize the geometry of their images as subsets of Euclidean space. We begin with a focus on signal data.

3.2 Preliminaries

We emphasize that we think about a patch, $\mathbf{x}_n$, in several different ways. Originally, $\mathbf{x}_n$ is simply a snippet of a time series. Then, we think about $\mathbf{x}_n$ as point in $\mathbb{R}^d$. Later, we also regard $\mathbf{x}_n$ as a vertex of a graph. A fourth perspective that is useful for our theoretical results defines the patch $\mathbf{x}_n$ as a point on a curve in $\mathbb{R}^d$, as we explain below. Keeping these four perspectives in mind is critical to our approach and understanding.

To clarify the fourth perspective, let $x(t)$ be a single-variable function.

Definition 5. *The **trajectory** associated with $x(t)$ is a vector valued function mapping $\mathbb{R} \to \mathbb{R}^d$ given by*

$$\mathbf{x}(t) = \left(x(t), x\left(t+\Delta t\right), x\left(t+2\Delta t\right), \ldots, x\left(t+(d-1)\Delta t\right)\right)^T. \tag{3.1}$$

*We refer to the parameter Δt as the **delay**.*

Now, assume that the function is sampled every $\tau > 0$ time units in order to produce the time series data

$$x_n = x(t_n), \qquad \text{where} \quad t_n = (n-1)\tau, \qquad \text{for } n = 1, 2, \ldots, N'.$$

When the sampling period τ is equal to the delay Δt, a point on the trajectory is equivalent to a patch, as defined in chapter 2:

$$\begin{aligned} \mathbf{x}(t_n) &= \big(x(t_n), x(t_{n+1}), \ldots, x(t_{n+(d-1)})\big)^T \\ &= \big(x_n, x_{n+1}, \ldots, x_{n+(d-1)}\big)^T, \qquad \text{for } n = 1, 2, \ldots, N. \end{aligned}$$

Therefore, when t is fixed, this correspondence allows us to identify a patch extracted from the signal at time t as $\mathbf{x}(t)$.

Before stating the main result of this chapter, we consider two motivating perspectives.

3.2.1 The patch-set and finite differences

It is clear that the values composing each patch can be used to approximate derivatives of the analog process up to order $d-1$ within an error that is on the order of the sampling period. For example, the forward difference approximation to the p^{th} derivative of $x(t)$ (provided it exists) can be written

$$\frac{1}{(\Delta t)^p} \sum_{k=0}^{p} (-1)^k \binom{p}{k} x(t + (p-k)\Delta t), \qquad \text{for } p \in \{1, 2, \ldots, d-1\}.$$

Equip $\mathbb{R}^d$ with rectangular coordinates $(q_1, q_2, \ldots, q_d)$, and let a *region of* $x(t)$ refer to the function $x(t)$ restricted to a *temporal neighborhood* $\{\tau \in \mathbb{R} : |t - \tau| < \epsilon, \epsilon > 0\}$ centered at t. If Δt is sufficiently small, and if a patch is extracted from a region of $x(t)$ in which $\left|\frac{d^p x}{dt^p}\right| \approx 0$, then it will lie in close proximity to the $(d-1)$-dimensional hyperplane

$$\sum_{k=0}^{p} (-1)^k \binom{p}{k} q_{k+1} = 0. \tag{3.2}$$

Furthermore, if we consider a patch extracted from a region of $x(t)$ such that

$$\left|\frac{d^k x}{dt^k}\right| < \epsilon_k, \qquad \text{for } k = 1, 2, \ldots, p \leq d-1, \tag{3.3}$$

then we can expect this patch will lie near to the intersection of p of the hyperplanes (3.2), provided Δt is small enough. As $\epsilon_k \to 0$, this set of inequalities corresponds to an intersection of p of the hyperplanes (3.2), resulting in a $(d-p)$ dimensional hyperplane in $\mathbb{R}^d$, further confining the space where we would expect patches capturing smooth changes in the time series to be located.

The previous perspective suggests that the organization of the patch set as points can encode derivative information of $x(t)$, and that patches leading to similar derivative approximations may be in close proximity in $\mathbb{R}^d$.

3.2.2 Patches as the pointwise image of linear operators

Note that the trajectory (3.1) can be regarded as the *pointwise image* of the function $x(t)$ mapped through a linear operator $\mathcal{T}$. More precisely, let Ω_0 and Ω_1 be two linear function spaces such that the set Ω_0 contains functions $x : \mathbb{R} \to \mathbb{R}$, while the set Ω_1 contains functions $\mathbf{x} : \mathbb{R} \to \mathbb{R}^d$. Define the map $\mathcal{T} : \Omega_0 \to \Omega_1$ by its action on $t \in \mathbb{R}$ as

$$\mathcal{T}x(t) = \mathbf{x}(t). \tag{3.4}$$

Clearly $\mathcal{T}$ is a linear operator between Ω_0 and Ω_1, so it is easy to characterize the image of $\mathcal{T}(\Omega_0)$ as a subspace of Ω_1.

With this in mind, consider the case that Ω_0 is the span of a single function. As a first example, let $\Omega_0 = \{\alpha \sin(\omega t) : \alpha \in \mathbb{R}\}$. Clearly the dimension of the image of $\mathcal{T}$ as a subspace of Ω_1 is at most one, according to the rank-nullity theorem. However, if we choose $d = 2$ and $\Delta t = \frac{\pi}{2\omega}$, then the pointwise image of $\mathcal{T}(\Omega_0)$ in $\mathbb{R}^2$ comprises concentric circles of radius α, occupying a subspace of dimension two. The situation is even more extreme if Ω_0 is the span of a piecewise constant function. For example, let Ω_0 be the span of the sign function

$$\operatorname{sgn}(t) = \begin{cases} -1 & \text{if } x \leq 0, \\ \\ 1 & \text{if } x > 0. \end{cases}$$

Then, for fixed $\alpha \neq 0$ and any value of d and Δt, the pointwise image of the sign function through $\mathcal{T}$ jumps between vertices of the d-dimensional hypercube $\mathbb{Q}^d = \{v \in \mathbb{R}^d : \|v\|_\infty = |\alpha|\}$. Such a

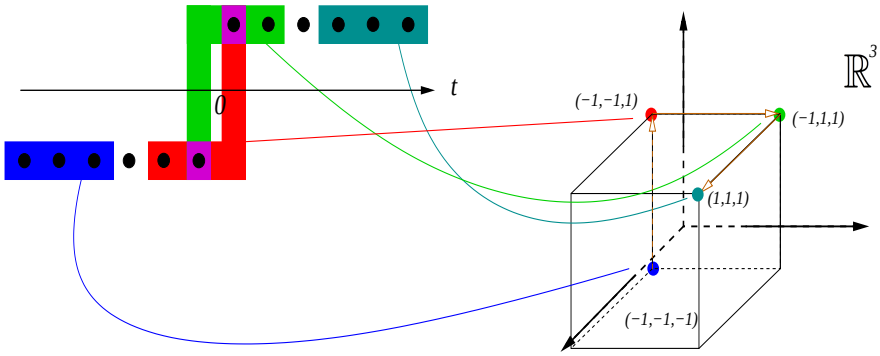

Figure 3.1: On the left we have the sign function with black dots representing a samples of the function that would compose patches. Different colors represent different patches of size $d = 3$. These patches are mapped to vertices of the cube on the right – a pointwise trajectory that occupies every distinct dimension of $\mathbb{R}^3$. This behavior persists as the patch size, d, increases.

pointwise image occupies every distinct dimension of $\mathbb{R}^d$ at some time t (see Figure 3.1). In other words, the pointwise image of $\mathcal{T}(\Omega_0)$ in $\mathbb{R}^d$ occupies a subspace whose dimensionality increases with the patch size, d

The examples above demonstrate that although Ω_0 and $\mathcal{T}(\Omega_0)$ may be simple in function spaces, the pointwise image of $\mathcal{T}(\Omega_0)$ in $\mathbb{R}^d$, in particular the patch-set generated by $x(t)$, is somewhat more obscure.

3.3 A lemma on the geometry of the patch-set in Euclidean space

The following lemma implies that there exist linear function spaces Ω_0 of dimension $p < d$ such that the pointwise image of $\mathcal{T}(\Omega_0)$ as a set in $\mathbb{R}^d$ is a p-dimensional subspace of $\mathbb{R}^d$, where $\mathcal{T}$ is defined in (3.4). Also, as described in appendix A.1, we can analytically compute this subspace given the functional form of Ω_0.

Lemma 1. *Assume that $x(t)$ is a solution to a p^{th}-order, linear, constant coefficient, homogeneous ordinary differential equation (ODE)*

$$\frac{d^p x}{dt^p} + c_{p-1}\frac{d^{(p-1)}x}{dt^{p-1}} + \cdots + c_1\frac{dx}{dt} + c_0 x = 0. \tag{3.5}$$

If this ODE has a characteristic equation with only simple roots, and $p < d$, then the trajectory

(3.1) is confined to a p-dimensional subspace of $\mathbb{R}^d$ that is uniquely characterized by the roots and the delay Δt.

Proof: *See appendix A.1.*

Remark: *If the ODE's characteristic equation has roots with multiplicity greater than one, then one can approximate that ODE's solutions with solutions to another ODE, whose coefficients are perturbed by an infinitesimally small amount. With probability one, this ODE has simple roots, and the previous lemma can be applied. The resulting approximation will be valid over a bounded interval of time. On this interval, we can use Corollary 3.7, presented below, in order to bound the deviation of the trajectories of the solutions associated with the original ODE.*

Lemma 1 implies that if $x(t)$ is the solution to (3.5), then the trajectory cannot be a "wild" curve, filling up all of $\mathbb{R}^d$, but is confined to a p-dimensional subspace. Furthermore, if d changes, the dimension of the subspace, p, remains constant. This fact can be used to test our presumptions about the data. Specifically, if we hypothesize that the signal is a solution to an ODE of the form (3.5), but we do not know the order p, then it would be possible to infer p by estimating the global dimensionality of the patch set using principal component analysis or singular value decomposition as d increases: the order p is the largest global dimensionality estimate obtained as d increases. Finally, Lemma 1 accounts for the discretization of the underlying scene, by essentially rotating the subspace in a way that depends on the delay Δt (see appendix A.1). Note that Lemma 1 only gives the maximum dimension of the subspace of $\mathbb{R}^d$ to which the trajectory belongs. So, it is possible that a function solving (3.5) could exhibit fewer than p degrees of freedom.

We mention two relevant corollaries of Lemma 1. Corollary 1 assumes that $x(t)$ is exactly a linear combination of distinct sinusoidal functions, while Corollary 2 assumes only that $x(t)$ can be approximated as a linear combination of solutions to (3.5) in order to bound the deviation of the trajectory (3.1) from a p-dimensional subspace of $\mathbb{R}^d$.

Corollary 1. *If $x(t)$ can be written as*

$$x(t) = \sum_{k=1}^{K} \Big(a_k \cos(\omega_k t) + b_k \sin(\omega_k t) \Big), \tag{3.6}$$

then the trajectory $\mathbf{x}(t)$ *is contained in a* $2K$*-dimensional subspace of* $\mathbb{R}^d$*.*

Proof: *See appendix A.1.1.*

Corollary 2. *Assume that* $x(t)$ *can be written as*

$$x(t) = \sum_{i=1}^{p} b_i \, y_i(t) + e(t), \qquad \text{for } t \in I \tag{3.7}$$

in some interval I *with positive length, where each* $y_i(t)$ *solves (3.5),* b_i *are expansion coefficients, and* $e(t)$ *is the error between* $x(t)$ *and the linear combination of* $y_i(t)$ *on* I*. It follows that if* $|e(t)| \leq \epsilon_1$ *for all* $t \in I$*, then the Euclidean distance between the trajectory segment* $\{\mathbf{x}(t) : t \in I\}$ *and the subspace containing the trajectories*

$$\mathbf{y}_i(t) = (y_i(t), y_i(t + \Delta t), \ldots, y_i(t + (d-1)\Delta t))^T, \qquad \text{for } i = 1, 2, \ldots, p$$

is at most $\sqrt{d}\,\epsilon_1$*.*

Proof: *See appendix A.1.2.*

As a consequence of Corollary 2, the patches extracted at times $t \in I$ will also lie at most $\sqrt{d}\,\epsilon_1$ away from a p-dimensional subspace of $\mathbb{R}^d$.

3.4 Generalization to images

In this section, we conjecture that Lemma 1 has a natural extension to image patches. We provide a sketch of a potential proof in appendix A.2, and describe the difficulty in completing this proof. In section 3.7, we provide empirical evidence which supports the conjecture.

We regard the image as a function $x(t, u)$, defined on (a subset of) $\mathbb{R}^2$. We think of an image

patch, $\mathbf{x}(t,u)$ as a d-by-d matrix:

$$\mathbf{x}(t,u) = \begin{pmatrix} x(t,u) & x(t,u+\Delta u) & \dots & x(t,u+(d-1)\Delta u) \\ x(t+\Delta t,u) & x(t+\Delta t,u+\Delta u) & \dots & x(t+\Delta t,u+(d-1)\Delta u) \\ & & & \\ x(t+(d-1)\Delta t,u) & x(t+(d-1)\Delta t,u+\Delta u) & \dots & x(t+(d-1)\Delta t,u+(d-1)\Delta u) \end{pmatrix}. \tag{3.8}$$

Furthermore, we assume that a local approximation to $x(t,u)$ can be written as

$$x(t,u) \approx X(t)Y(u) \qquad \text{for all } (t,u) \in \Omega, \tag{3.9}$$

where $\Omega \subset \mathbb{R}^2$.

Conjecture 1. *Assume that $X(t)$ is a solution to a p^{th}-order, linear, constant coefficient, homogeneous ordinary differential equation*

$$\frac{d^pX}{dt^p} + a_{p-1}\frac{d^{(p-1)}X}{dt^{p-1}} + \cdots + a_1\frac{dX}{dt} + a_0X = 0. \tag{3.10}$$

Also assume that $Y(u)$ is a solution to another q^{th}-order, linear, constant coefficient, homogeneous ordinary differential equation

$$\frac{d^qY}{du^q} + b_{m-1}\frac{d^{(q-1)}Y}{du^{q-1}} + \cdots + b_1\frac{dY}{du} + b_0Y = 0. \tag{3.11}$$

If the ODEs (3.10) and (3.11) have characteristic equations with only simple roots, and $p, q < d$, then the patch space generated by (3.8) for all $(t + k\Delta t, u + l\Delta u) \in \Omega$ and $k, l = 0, 1, \dots, d-1$ is confined to a pq-dimensional subspace of $\mathbb{R}^{d^2}$ that is uniquely characterized by the constant coefficients of the linear differential equations.

Remark: *See appendix A.2 for the beginning of a possible proof.*

Although the assumption (3.9) is somewhat limiting, in some cases, we can still account for more complex content in an image patch. More precisely, in Figure 3.2 we show the approximations

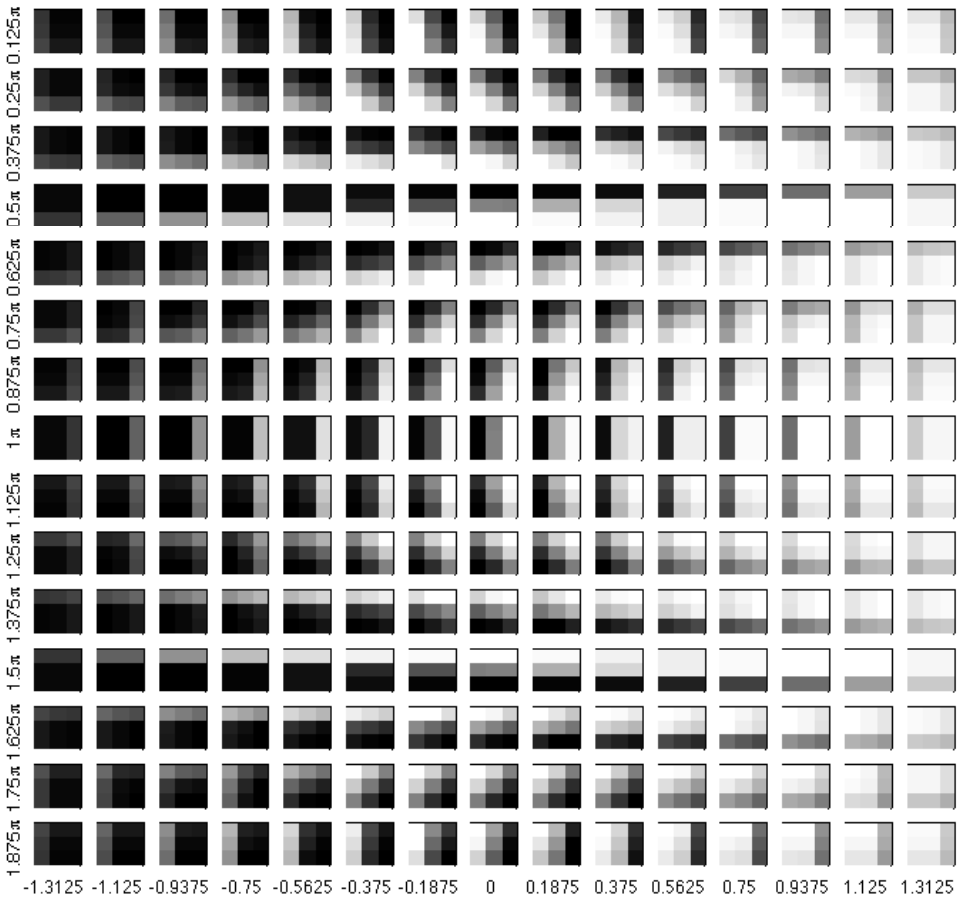

Figure 3.2: Approximations to the patch-set of Figure 2.4. Each approximation lives in a four-dimensional subspace of $\mathbb{R}^d$.

to the ideal patch set of Figure 2.4. The approximations in Figure 3.2 are obtained by projecting the patches of Figure 2.4 onto the span of only two Fourier basis vectors. Therefore, according to the conjecture 1, these approximations live in a four-dimensional subspace of $\mathbb{R}^d$. Given the assumption (3.9), it is expected that when the step edge inside a patch is oriented vertically or horizontally, the approximations are more accurate. Thus, in order to more accurately approximate a step edge that is not aligned vertically or horizontally, we can rotate the vertical and horizontal directions inside a patch. Then, we can infer that even these patches live close to a four-dimensional

subspace. The union of these subspaces may help characterize the nonlinear structure in the ideal patch-set.

3.5 Normalization

In many applications, we may want to identify two patches from a signal or image as similar even though they may be different in their amplitudes or in their averages/ offsets from zero. To ensure that these patches are connected, we modify the metric used to create the patch graph in (2.3). This modification of the metric is equivalent to normalizing the patches. Below we describe two normalizations and the advantages of each. We note that the dimensionality estimates of sections 3.3 and 3.4 are still upper bounds on the ambient dimensionality of the trajectory, even after normalizing.

3.5.1 Removing the mean from each patch

To remove the mean from a patch $\mathbf{x}(t)$, at a fixed time $t \in \mathbb{R}$, we first compute the mean of $x(t)$ over the interval $[t, t + d\Delta t)$, given by $\overline{x}(t) = d^{-1} \sum_{k=0}^{d-1} x(t + k\Delta t)$. Then, we compute the centered patch

$$\mathbf{x}_0(t) = \big(x(t) - \overline{x}(t), \ldots, x(t + (d-1)\Delta t) - \overline{x}(t)\big)^T. \tag{3.12}$$

This procedure effectively estimates and removes a slowly varying drift by computing a running average over the entire signal.

Geometrically, the normalized patch (3.12) lies on a curve on a hyperplane in $\mathbb{R}^d$. Indeed, after subtracting the mean, the patch lies on the hyperplane of $\mathbb{R}^d$ defined by $\sum_{n=1}^{d} x_n = 0$. In addition, as shown in appendix A.3, the Euclidean norm of the normalized patch (3.12) is equal to the distance between the original trajectory and the subspace spanned by the constant vector $\mathbf{1} = (1, 1, \ldots, 1)^T \in \mathbb{R}^d$. This is useful because the distance between the trajectory and the subspace spanned by the constant vector is related to the *local-mean-oscillation* of the signal, as shown in Section A.3. The signal's local-mean-oscillation quantifies how much it deviates from its average on a local interval, and has important applications in functional analysis [41, 45].

Note that the normalization (3.12) tends to preserve the high-frequency content that exists in the original signal or image, as shown in appendix A.4,

Finally, constructing the patch-graph using the normalized patches (3.12), is equivalent to using the using the metric

$$\rho_0(\mathbf{x}_n, \mathbf{x}_m) = \left\| (\mathbf{x}_n - d^{-1}\langle \mathbf{x}_n, \mathbf{1}\rangle \mathbf{1}) - (\mathbf{x}_m - d^{-1}\langle \mathbf{x}_m, \mathbf{1}\rangle \mathbf{1}) \right\|$$

in the definition of the edge weights, given in (2.3).

3.5.2 Unifying the Euclidean norm of each patch

To make all patches have the same Euclidean norm, we project the patch $\mathbf{x}(t)$ onto the unit sphere and define the normalized patch

$$\mathbf{x}_1(t) = \frac{\mathbf{x}(t)}{\|\mathbf{x}(t)\|}. \tag{3.13}$$

The normalized patch (3.13) characterizes the local oscillation of $x(t)$ in a manner that is independent of changes in amplitude. Geometrically, the normalized patch (3.13) lies on a curve on the $d-1$ dimensional unit sphere in $\mathbb{R}^d$.

Note that the normalization (3.13) also preserves frequency content when the signal is sufficiently smooth enough, as discussed in appendix A.4.

Finally, constructing the patch-graph using the normalized patches (3.13), is equivalent to using the using the metric

$$\rho_1(\mathbf{x}_n, \mathbf{x}_m) = \left\| \frac{\mathbf{x}_n}{\|\mathbf{x}_n\|} - \frac{\mathbf{x}_m}{\|\mathbf{x}_m\|} \right\| \tag{3.14}$$

in the definition of the edge weights, given in (2.3). The metric ρ_1 is useful because it is invariant under any global scaling of the signal, and it is not sensitive to changes in the local energy of the signal (as measured by the Euclidean norm of the difference between patches). In addition, the metric ρ_1 allows us to specifically detect changes in the signal's local frequency content, or local smoothness.

3.6 Lemma 1 applied

In this section, we consider some applications of Lemma 1.

3.6.1 Sparse representation of an ODE's patch-set

In this section, we consider the patch-set extracted from a solution to an ODE of the form (3.5). We demonstrate that the patch-set is contained in the subspace predicted by Lemma 1 by comparing two approximations to the patch's content. One approximation is obtained with vectors that span the two-dimensional subspace given in the proof of Lemma 1. The other approximation minimizes the Euclidean norm of the error between the true patch and the approximation when using only three elements from a Fourier basis. This comparison demonstrates the utility of having a functional form of the subspace predicted by Lemma 1 over a simple Fourier analysis of the patch-set.

Figure 3.3 plots solutions to the second-order ODE $x''+\omega^2 x = 0$ for $2\pi \leq \omega \leq 4\pi$ for $t \in [0, 1)$ as a black curve with intial conditions $x(0) = -1$ and $x'(0) = 1$. For a fixed value of ω, we create only one patch in $\mathbb{R}^{25}$ by sampling the ODE's solution 25 times for $t \in [0, 1)$. The red circles represent the best nonlinear approximation to the data using three elements from a Fourier basis. More precisely, components of a patch extracted from the solution to the ODE can be written as

$$x_j = \sum_{k=1}^{\lceil d/2 \rceil} \alpha_k \cos\big(2\pi(k-1)\,(j-1)/d\big) + \beta_k \sin\big(2\pi(k-1)\,(j-1)/d\big), \quad \text{for } j = 1, 2, \ldots, d, \quad (3.15)$$

where α_k and β_k are chosen to satisfy the initial conditions.

We define the components of the best nonlinear approximation using the Fourier basis as

$$\tilde{x}_j = \sum_{l \in \mathcal{A}} \alpha_l \cos\big(2\pi(l-1)\,(j-1)/d\big) + \beta_l \sin\big(2\pi(l-1)\,(j-1)/d\big), \quad \text{for } j = 1, 2, \ldots, d,$$

where $\mathcal{A} \subset \{1, 2, \ldots, 11, 12 = \lceil d/2 \rceil\}$ is the set of indices associated with the three expansion coefficients α_k with the largest magnitude in (3.15).

As expected, when the patch data is not periodic on the domain of the patch, the Fourier approximation suffers. On the contrary, the reconstruction to the patch data using vectors which

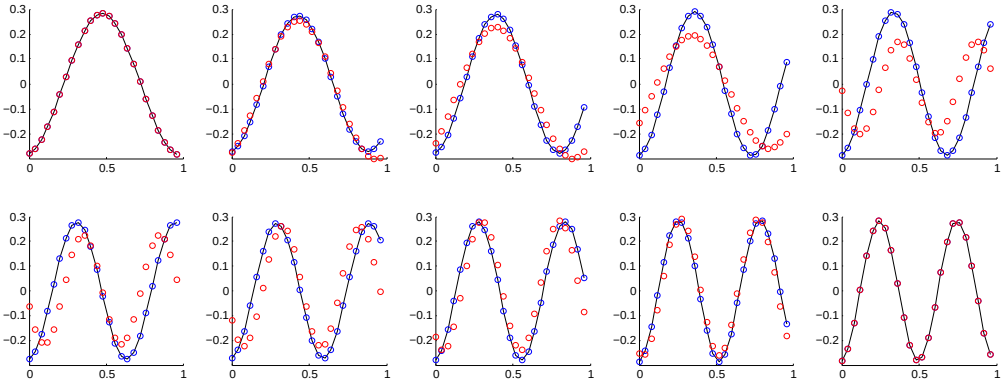

Figure 3.3: We represent the 25 time-samples composing a patch extracted from a signal $\alpha\cos(\omega t)+\beta\sin(\omega t)$ as a graph over $[0,1)$ with a black curve. Each plot corresponds to a different value of ω. The red circles represent the best nonlinear approximation to the patch data using a Fourier basis. The blue circles represent the approximation to the patch data using vectors which span the two-dimensional subspace predicted by Lemma 1.

span the two-dimensional subspace predicted by Lemma 1 is exact, as supported by the blue circles in Figure 3.3.

3.6.2 The patch-set of a linear chirp

Consider the linear chirp $x(t) = \cos\left(\frac{\omega}{2}t^2\right)$ for times $t \in [0, 2\pi]$. Notice that at each time $t_0 \in [0, 2\pi]$, the chirp can be approximated with a cosine of frequency ωt_0:

$$x(t) \approx \cos\left(\omega t_0\left(t - \frac{t_0}{2}\right)\right), \tag{3.16}$$

for all t within a sufficiently small neighborhood of t_0. We plot the linear chirp $x(t) = \cos\left(\frac{\omega}{2}t^2\right)$ in Figure 3.4. We fix $\omega = 2$, $d = 3$, and $\Delta t = \frac{\pi}{50}$. Notice that the approximation to the chirp (3.16) is a sinusoid whose frequency, ωt_0, increases linearly with the time t_0 about which we approximate $x(t)$. At each time t_0, we can identify the two-dimensional subspace of $\mathbb{R}^d$ that confines solutions to the ODE

$$x'' + (\omega t_0)x = 0. \tag{3.17}$$

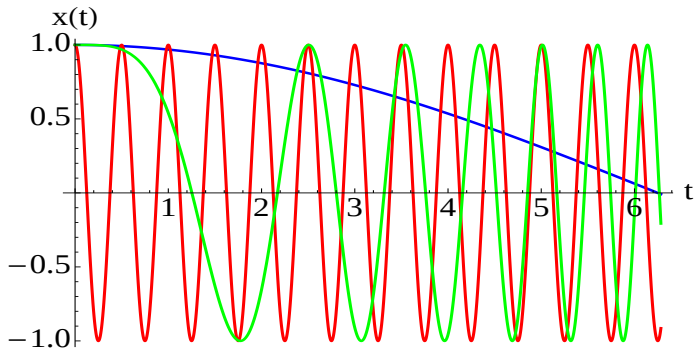

Figure 3.4: Two sinusoidal functions in red and blue, and a linear chirp whose frequency content varies between the frequencies of the sinusoidal functions in green.

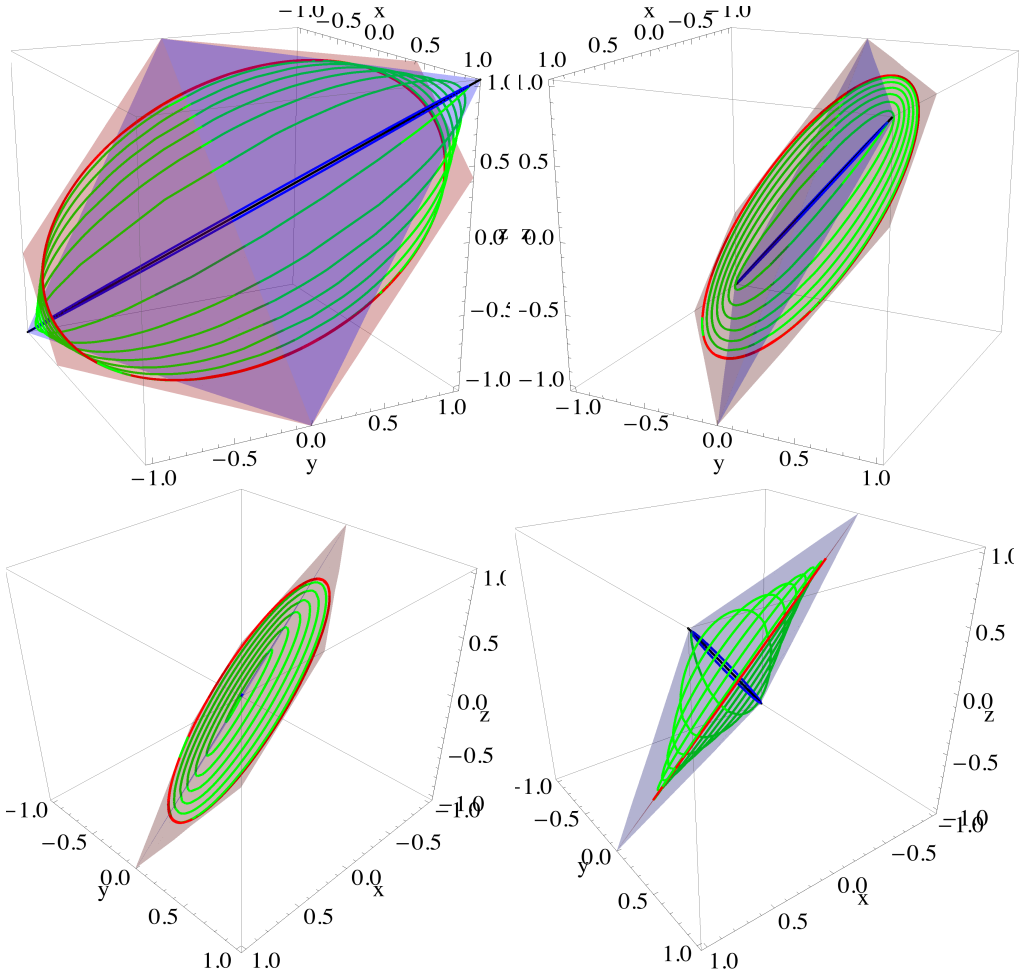

Figure 3.5: Several different views of the curves associated with a linear chirp (in green), and two cosines waves oscillating at the chirp's minimum and maximum frequencies (in blue and red, respectively).

Thus, according to Lemma 1, the trajectory will begin at time $t = 0$ near a subspace associated with the frequency $\omega_{min} = \omega(d-1)\Delta t$. Then, at time $t = 2\pi$, the trajectory associated with the chirp will end near the subspace associated with the frequency $\omega_{max} = \omega(2\pi + (d-1)\Delta t)$ at time $t = 2\pi$. Confirming our expectations, Figure 3.5 shows a green curve (representing the chirp's trajectory) that wanders between two ellipses. Each ellipse represents the trajectory associated with a solution to the ODE (3.17). The red and blue colors correspond to the frequencies ω_{min} and ω_{max}, respectively.

3.7 A first look at the patch-set

The goal of this section is to provide the reader with some intuition about the geometry of the patch-set and the associated patch-graph. This will help us motivate our graph models and the analysis of their geometries, which are presented in chapter 4.

3.7.1 Examples of signals and images

We construct the patch-set associated with some examples of signals and images. Because it is not practical to visualize the patch-set in $\mathbb{R}^d$ when $d = 25$, we display the projection of the patch-set onto the three-dimensional space that captures the largest variance in the patch-set (computed using principal component analysis). Figure 3.6 displays three signals $\{x_n\}, n = 1, \ldots, N'$, with $N' = 2072$. Patches of size $d = 25$ samples are extracted around each time sample, which results in the maximum overlap between patches. Signal A is a chirp, signal B is a row of the image Lenna (shown in Fig. 3.7-D), and signal C is a seismogram [84].

In order to quantify the local regularity of signals A and B, we compute the variance over each patch, and color the curve according to the magnitude of the local variance: hot (red) for large variance and cold (blue) for low variance. The color of signal C encodes the temporal proximity to the arrival of a seismic wave associated with an earthquake: hot color indicates close proximity, while cold color corresponds to baseline activity. Identifying arrival-times is necessary for purposes such as locating an earthquake's epicenter. This example illustrates the application of the present

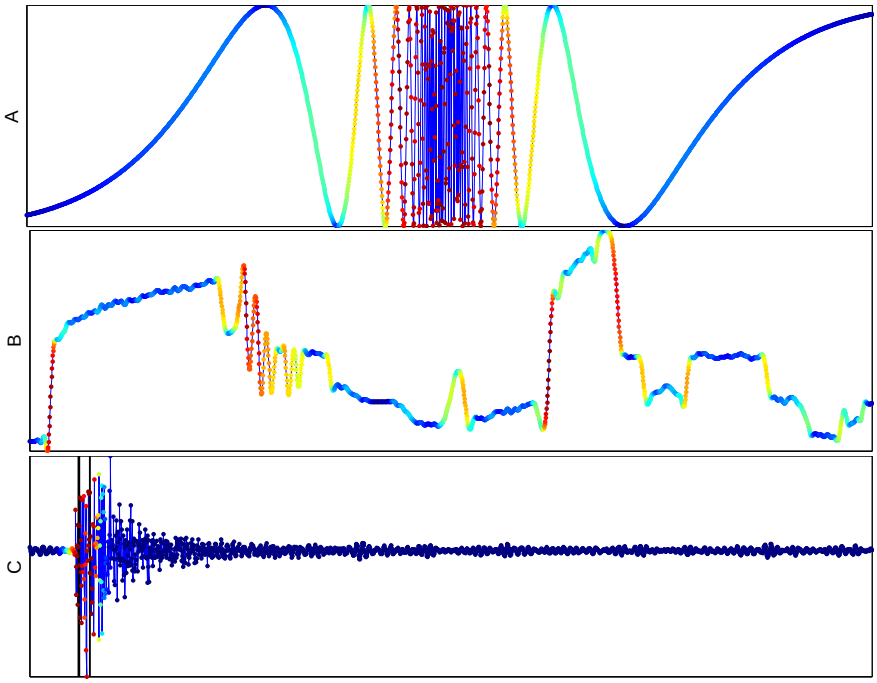

Figure 3.6: A, B, C: time series composed of $N' = 2072$ samples. The color of signals A and B encodes the local variance (large = red, low = blue). C: seismogram; the color indicates the temporal proximity to a seismic arrival, identified by vertical black lines. See text for more details.

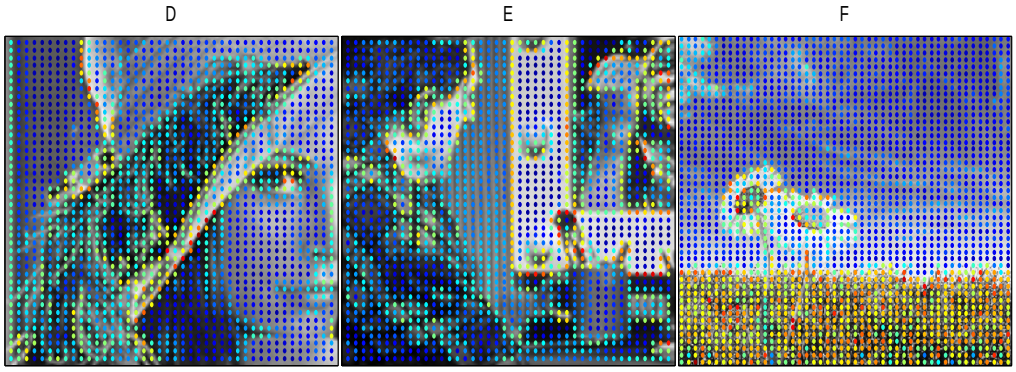

Figure 3.7: D, E, and F: image of size 128×128, 128×128, and 240×240 pixels, respectively. The color of the pixel at the center of each patch encodes the local variance of the image's intensity.

work to the problem of detecting seismic waves [84].

Figure 3.7 displays three images. We extract patches of size 5×5. Here, the patches are not maximally overlapping: we collect every third patch in the horizontal and vertical directions for images D and E, while we collect every fifth patch in each direction for image F. This results in patch-sets of size 42×42 for images D and E, and of size 48×48 for image F. As before, the color of a pixel in the images encodes the local variance within the patch centered at that pixel.

3.7.2 Projections of the patch-sets

Figure 3.8 shows the projections of each of the six patch-sets. Distances in Figure 3.8 correspond to the normalized distance ρ_1, defined in (3.14). We observe that patches with high variance (red-orange) appear to be scattered all over $\mathbb{R}^d$. These patches correspond to regions where the image intensity varies rapidly. Patches with low variance (blue-green), which correspond to regions where the signal is smooth and varies very little, tend to be concentrated along one-dimensional curves (for time series) and two-dimensional surfaces (for images). These visual observations can be confirmed when computing the actual mutual distances between patches (data not shown).

The organization of the patches in the patch-set can be explained using simple arguments. Let us assume that the sequence $\{x_n\}$ corresponds to the sampling of an underlying differentiable function $x(t)$, and assume that the derivative of $x(t)$, given by $x'(t)$, remains small over the interval of interest. In this case, if two patches $\mathbf{x}_n$ and $\mathbf{x}_m$ overlap significantly – i.e. $|n - m|$ is small – then they will be close to one another in $\mathbb{R}^d$. Indeed, the values of the coordinates of patches $\mathbf{x}_n$ and $\mathbf{x}_m$ will be very similar, since the signal $x(t)$ varies slowly. In principle, if the sampling is fast enough, the patches should lie along a one-dimensional curve in $\mathbb{R}^d$. By the same argument, when $x(t)$ exhibits rapid changes, the magnitude of the derivative, $|x'(t)|$, can be very large, and therefore temporally neighboring patches are not guaranteed to be spatial neighbors in $\mathbb{R}^d$. This argument allows us to understand the distribution of the patches in the signal B, or the image F.

Instead of characterizing patches according to the local smoothness of the underlying function, we can also analyze the distribution of the patches according to the function's local frequency

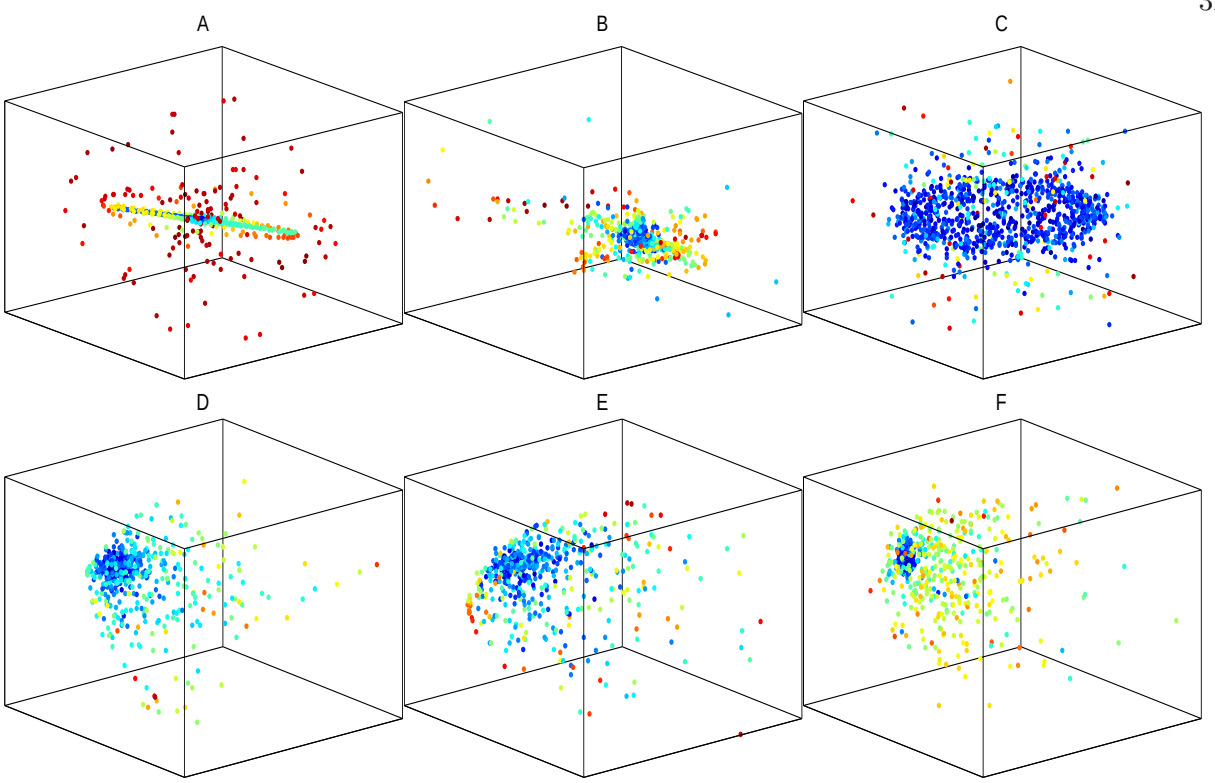

Figure 3.8: Principal component analysis of patch-sets associated with the time series A-C and the images D-F. Each point represents a patch; the color encodes the variance within the patch (see Figures 3.6 and 3.7.)

information. This will help us understand the structure of the patch-set for signal A. For this type of signal, it is appropriate to measure the distance between the normalized patches, $\mathbf{x}_n/\|\mathbf{x}_n\|$ and $\mathbf{x}_m/\|\mathbf{x}_m\|$ after computing the Fourier transforms (a simple rotation of $\mathbb{R}^d$) of the respective patches. This process is akin to the concept of time-frequency analysis. We expect that regions of the signal with little local frequency changes will cluster in $\mathbb{R}^d$: this is the case for the blue patches of the chirp signal A. On the contrary, when the local frequency content changes rapidly (as in the middle of the chirp signal A), the corresponding patches will be far away from one another in $\mathbb{R}^d$: this is the case for the red patches of signal A (see Figure 3.8-A).

We can also try to understand the organization of the patch-set for the seismogram C. Let us assume that $\{x_n\}$ is obtained by sampling a function of the form $x(t) = b(t) + w(t)$, where $w(t)$ represents a seismic wave and $b(t)$ represents baseline activity. We can expect that $w(t)$ is a rapidly

oscillating transient with rich frequency content, while $b(t)$ is varying slowly. Now consider two patches $\mathbf{x}_n$ and $\mathbf{x}_m$. It can be shown that if both patches $\mathbf{x}_n$ and $\mathbf{x}_m$ are extracted from the baseline function, $b(t)$, and do not contain any part of the energetic transient, then their mutual distance is expected to be small. In addition, if $\mathbf{x}_n$ contains part of the energetic transient $w(t)$ and $\mathbf{x}_m$ is extracted from the baseline $b(t)$, then their mutual distance is expected to be large. Finally, if $\mathbf{x}_n$ and $\mathbf{x}_m$ are composed of two different parts of $w(t)$, then their mutual distance is also expected to be large (provided the patches are sufficiently long and $w(t)$ oscillates sufficiently fast). More generally, one can expect that two patches extracted from two different energetic transients $w_1(t)$ and $w_2(t)$ will be at a large distance from one another (see section 5.2 and [84]).

Finally, Lemma (1) can also be used to interpret the inter-patch distances. In particular, portions of the signal that can be well-approximated using few ODE solutions will generate patches that live in a smaller region of $\mathbb{R}^d$ (even after normalizing) than those patches generated by portions of the signal that require very many ODE solutions to be approximated.

3.7.3 Local dimensionality estimates via local PCA

To estimate the local dimensionality of the patch-set, we perform local PCA at each patch in $\mathbb{R}^d$, and record the minimum number of components required to capture at least 90% of the local neighborhood's variation. More precisely, we fix a positive integer ν. For each patch, $\mathbf{x}_n$, we determine its ν nearest neighbors in $\mathbb{R}^d$ using the ambient, Euclidean norm. Refer to the neighboring patches of $\mathbf{x}_n$ as $\mathbf{x}^{n,1}, \mathbf{x}^{n,2}, \ldots, \mathbf{x}^{n,\nu}$, and organize the patches as rows of the ν-by-d matrix $\mathbf{X}_n$. Let $\boldsymbol{\nu}$ be the d-by-d constant matrix with every entry equal to ν^{-1}, and form the ν-by-d matrix $\mathbf{Y}_n = (\mathbf{I} - \boldsymbol{\nu})\mathbf{X}_n$. One can verify that the ith row of $\mathbf{Y}_n$ is simply the patch $\mathbf{x}^{n,i}$ with its mean removed. For $k = 1, 2, \ldots, d$, let $\mathbf{p}_k$ denote the unit-norm eigenvector of the covariance matrix $(\nu - 1)^{-1}\mathbf{Y}_n^T\mathbf{Y}_n$ associated with the kth largest eigenvalue. We form the d-by-l matrix $\mathbf{P}_l$, using $\mathbf{p}_1, \ldots, \mathbf{p}_l$ as columns. The matrix $\mathbf{P}_l\mathbf{P}_l^T$ is the orthogonal projection onto the subspace of $\mathbb{R}^d$ spanned by $\mathbf{p}_1, \ldots, \mathbf{p}_l$. The norm

$$\varepsilon(l, n) = \|(\mathbf{I} - \mathbf{P}_l\mathbf{P}_l^T)\mathbf{X}_n\|, \tag{3.18}$$

where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^d$, measures the error involved in approximating the neighborhood of $\mathbf{x}_n$ with a subspace of dimension $l+1$ (we add one for the constant component). In addition (3.18) decreases as l increases. Therefore, we estimate the local dimensionality of the patch-set around $\mathbf{x}_n$ in $\mathbb{R}^d$ as the minimum l such that $\varepsilon(l,n) < 0.1 \max_m[\varepsilon(1,m)]$.

We show dimensionality estimates for the example time series and images A-F in Figures 3.9 and 3.10. Notice that near the center of signal A, where aliasing is present, we have a low dimensionality estimate. This can be understood as follows: The patch comprises $d = 25$ samples, and so at the level of the patch, we can only see at most 12 full oscillations on the patch. Therefore, when the signal oscillates faster than $\lfloor d/2 \rfloor$, the content of the signal will alias to a lower frequency, and appear smoother, thereby requiring fewer sinusoids to approximate locally, and thus a lower dimensionality upper bound according to Lemma (1). For comparison, consider the dimensionality estimate associated with signal C. Although there is likely aliasing taking place with signal C, there is still enough broad-band energy — meaning both low and high frequencies are present in the signal — to cause a large local dimensionality estimate.

The image patch data also exhibits low local dimensionality estimates in smooth regions of the image itself, as illustrated in Figure 3.9. Analogous to the aliasing in signal A, observe that there are also low dimensionality estimates corresponding to patches extracted from the flannel shirt in image E or the grass in image F.

3.7.4 From the patch-set to the patch-graph: the weight matrix W

Having gained some understanding about the organization of the patch-set, we now move to the structure of the patch-graph and its weight matrix $\mathbf{W}$. Figure 3.11 displays the weight matrices built from the patch-sets that correspond to the time series A-C (top) and the images D-F (bottom). Note that when processing time series A-C, the columns (or equivalently, the rows) of $\mathbf{W}$ can be identified with temporally-ordered time-samples. Therefore, a large main diagonal in the weight matrix correspond to patches that are close in time and also close in $\mathbb{R}^d$. For instance, consider the time series A and its associated weight matrix. The dark bands near the top-left

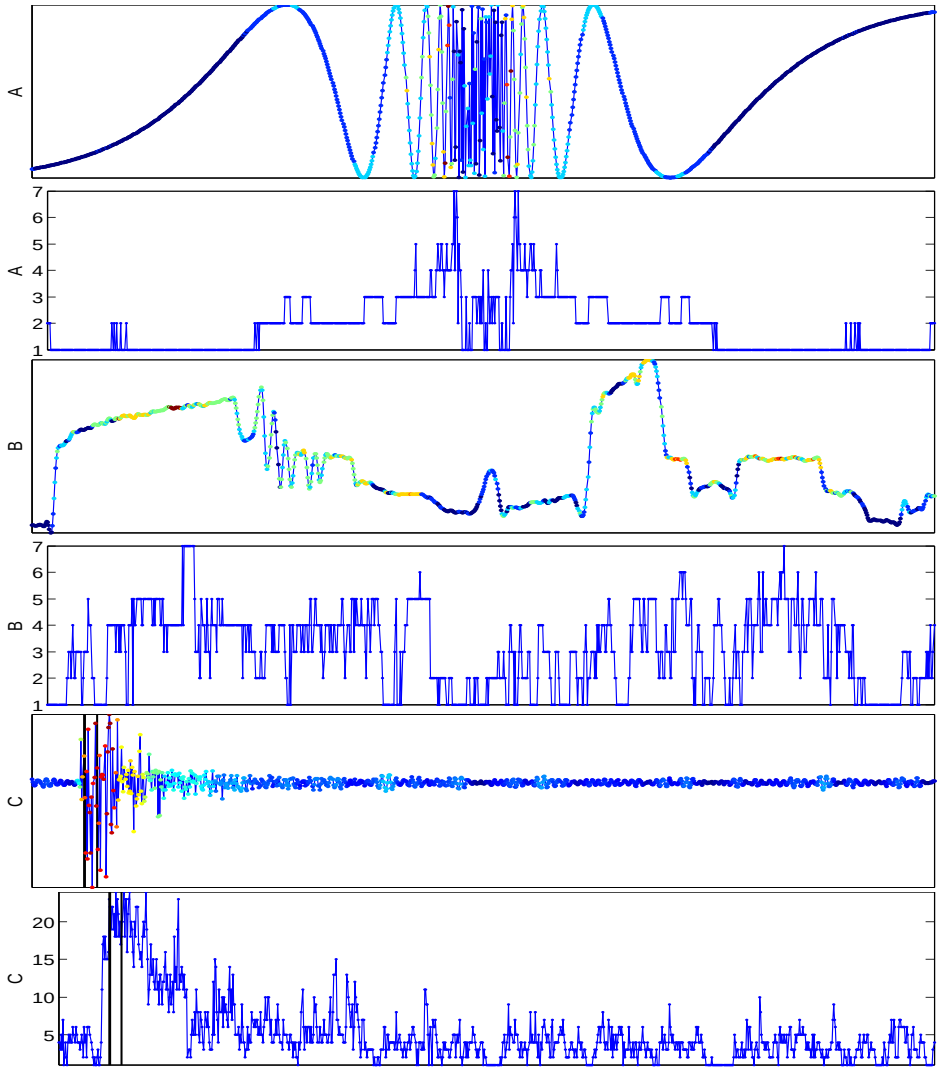

Figure 3.9: Local dimensionality estimates of the patch-set associated with time series A-C as points in $\mathbb{R}^{25}$. For A, B, and C, we plot both dimensionality estimates and the signal, with time samples color coded according to the estimated local dimensionality.

and bottom-right of the diagonal correspond to the slowly varying oscillations near the beginning and end of the chirp (see Figure 3.6). Indeed, large entries near the diagonal of $\mathbf{W}$ is a direct consequence of relatively little variation in the time series. On the other hand, the columns of $\mathbf{W}$ corresponding to portions of the time series that exhibit rapid local changes (center of Figure 3.11-A) tend to lack such prominent diagonal structures. For such regions of the matrix $\mathbf{W}$, the entries are no longer concentrated along the diagonal, and are shattered across all rows and columns

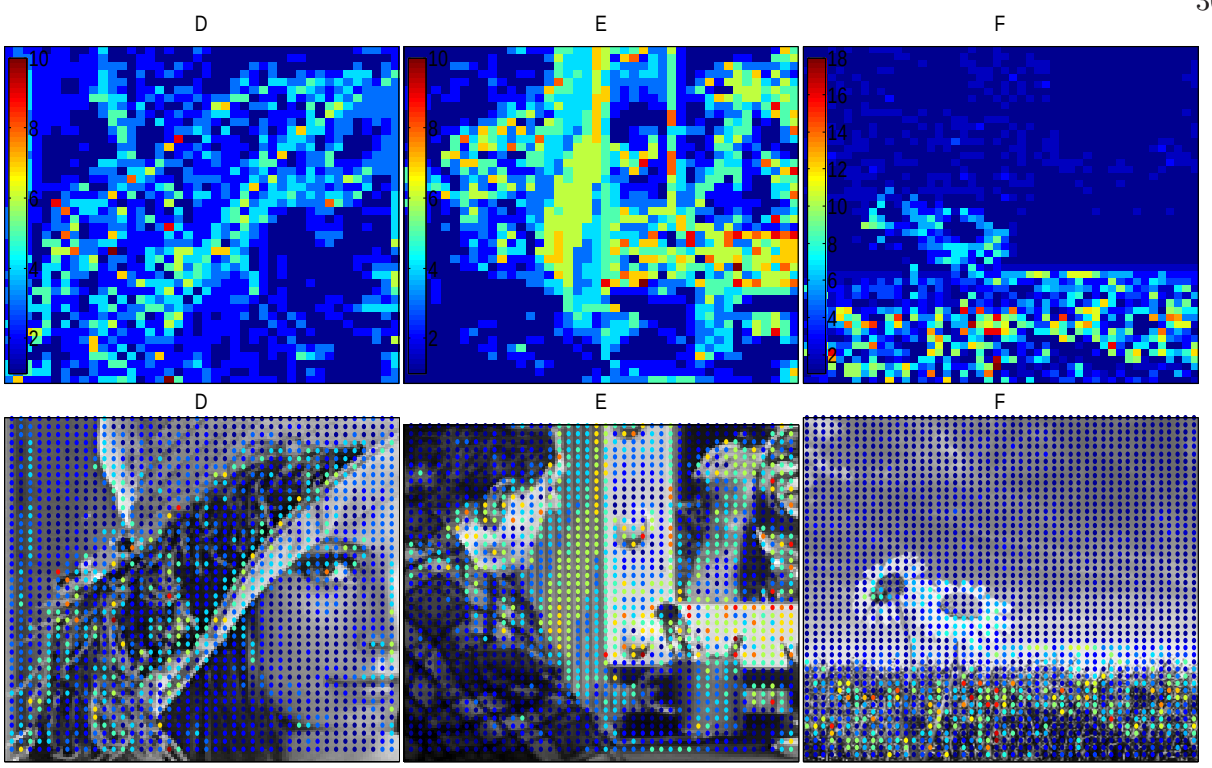

Figure 3.10: Local dimensionality estimates of the patch-set associated with images D-F as points in $\mathbb{R}^{25}$. The top row shows the estimated dimensionality as a colored square overlaid on the image plane, note the color bar. The bottom row shows estimates overlaid on the original image for comparison.

(see the center of $\mathbf{W}$ in Figure 3.11-A; the columns correspond to the fastest oscillations at the center of the chirp). The large distances between these patches are also apparent in the lighter pixel intensities, representing relatively smaller edge-weights. Note that the patches extracted from the seismic data are very far apart, as indicated by the much lighter shades of gray. It is more difficult to relate the ordering of an image's weight matrix to locations in the image itself. For the weight matrices associated with images D-F, the ordering of the columns is equivalent to the order in which the patches were collected from the image plane: first left-to-right, then top-to-bottom (similar to a raster scan, or how one would read pages of a book). Hence, periodically repeating dark blocks in the weight matrices associated with images D-F are indicative of image patches that are close in $\mathbb{R}^d$ and close in the image-plane as a result of relatively little change in the image's

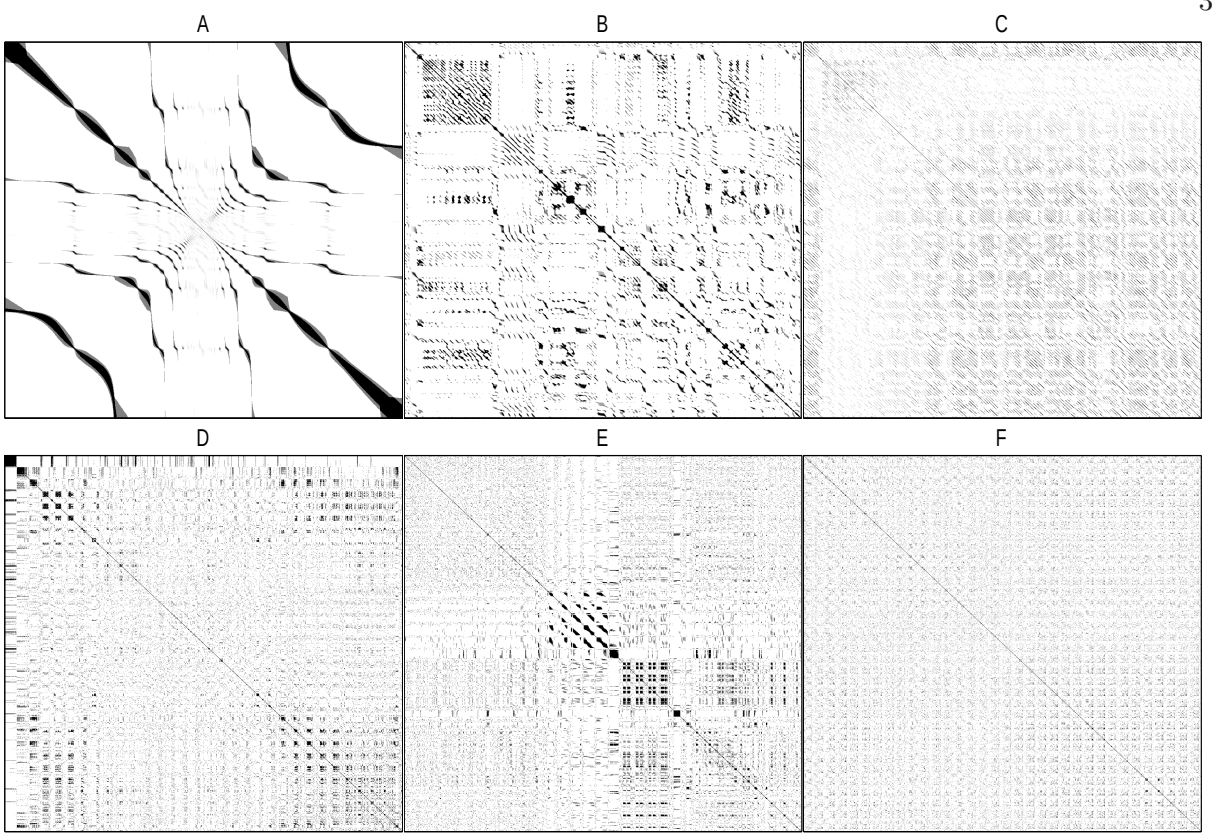

Figure 3.11: The weight matrices $\mathbf{W}$ associated with signals A-F are displayed as images: $w_{n,m}$ is encoded as a grayscale value: from white ($w_{n,m} = 0$) to black ($w_{n,m} = 1$). Dark structures along the diagonal of the $\mathbf{W}$ matrix associated with the time series A-C indicate that patches that are close in time are also close in $\mathbb{R}^d$.

local content. For example, the dark square-like structure that appears near the main diagonal of $\mathbf{W}$ in Figure 3.11-E, and which spans roughly one fifth of the number of columns, corresponds to the mirror's smooth, light border in image E.

Of course, a permutation of the patch indices might obfuscate the aforementioned structure in the weight matrix. In this case, the graph's parametrization and our theoretical conclusions are unaffected. We simply assume order in the patch index relative to the signal domain so that we can visually interpret the structure in $\mathbf{W}$.

3.8 Conclusion

The experiments in section 3.7 highlight the fact that regions of an image, or of a signal, that contain anomalies (e.g. singularities, edges, rapid changes in the frequency content, etc.) are scattered all over the patch-set, making their detection and identification extremely difficult (see Figures 3.8, 3.9, and 3.10). In particular, patches extracted from a portion of a signal or image with smooth content will require few ODE solutions to be well-approximated locally, and thus will lie close to a low-dimensional subspace of $\mathbb{R}^d$, according to Corollary 2. On the other hand, patches with sharp edges, or other drastic, localized changes will likely require many constituent ODE solutions to be equally well-approximated, and therefore these patches will occupy a relatively high-dimensional region of $\mathbb{R}^d$. Because the anomalous patches are usually the most interesting ones, we need to find a new parametrization of the patch-set that concentrates the anomalies and separates them from the smooth baseline part of the image. The structure of $\mathbf{W}$ in the "rough regions" suggests that patches that contain anomalies appear to be very well connected (see the center of Figure 3.11-D, which corresponds to the boa on the hat of Lenna). This concept can be quantified by studying how fast a random walk would reach all patches in these rough regions, and suggests that we should consider studying the *hitting times* associated with a random walk on the patch-graph. In the next chapter we formalize this concept and propose a parametrization of the patch-set in terms of the commute time on the patch-graph. A theoretical analysis of this approach is provided in section 4.4

Chapter 4

Parametrizing the patch-graph

4.1 Introduction

In this chapter, we analyze the effect that the organization of patches has on the embedding of the patch-graph based on the commute time metric. The commute time and its relation to other diffusion-based metrics are given in section 4.2. Our main result on the embedding of graphs that model general patch-graphs is described in section 4.4. We test our theoretical conclusions on the graph models using signal data in section 4.5.

4.1.1 The fast and slow patches

We first introduce the concept of *fast* and *slow* patches. We have noticed that patches that contain anomalies (discontinuities, edges, fast changes in frequency, etc.) in the original signal lead to regions of the matrix $\mathbf{W}$ where the nonzero entries are scattered all around. We call such patches *fast patches* because, as we will see in the following, a random walk will diffuse extremely fast in such regions of the patch-graph. Conversely smooth regions of the signal lead to *slow patches* that are associated with a small number of large entries in $\mathbf{W}$, which are concentrated near the diagonal. We will see that a random walk initialized in the slow patch region of the patch-graph will diffuse very slowly.

4.2 A better metric on the graph: the commute time

As explained in section 3.8, we propose to replace the Euclidean distance, which leads to the scattering of the fast patches seen in Figure 3.8 by a notion of distance that quantifies the speed at which a random walk diffuses on the patch-graph. We propose to use the commute time. Parametrizing the graph using its commute time distance is closely related to parametrizing the graph using its diffusion distance [23] (see Section 4.2.2). Although the works [16, 80, 81] do not explicitly embed vertices of the patch-graph based on the diffusion distance, they also study a random walk on the patch-graph, and define the diffusion distance in terms of this walk. In these studies, noise is removed by evolving the diffusion process for a small time. A detailed comparison of our approach with the seminal work of [80] is provided in section 7.3. We note that the notion of the first-passage time associated with a diffusion (which is equivalent to the hitting time associated with a random walk) has been used extensively to characterize the geometry of complex networks, and random media (e.g. [9, 25] and references therein). It is therefore natural to analyze the patch-graph (and inherently the patch-set) with this distance.

4.2.1 A random walk on the patch-graph

In order to define the commute time between two vertices, we first need to define a random walk on the graph. In our problem, the random will lead to a notion of global proximity between patches. We consider a first-order homogeneous Markov process, Z_k, defined on the vertices of the patch-graph, Γ, and evolving with the transition probability matrix $\mathbf{P}$ given by

$$\mathbf{P}_{n,m} = \text{Prob}(Z_{k+1} = \mathbf{x}_m | Z_k = \mathbf{x}_n) \triangleq \frac{w_{n,m}}{\sum_l w_{n,l}} = \frac{\mathbf{W}_{n,m}}{\mathbf{D}_{n,n}}. \tag{4.1}$$

Consider a slow patch $\mathbf{x}_n$ extracted from a regular/smooth part of the signal. If the random walk starts at $\mathbf{x}_n$, then it can only travel along the low-dimensional structure that corresponds to the temporal neighbors of $\mathbf{x}_n$ (see e.g. Figure 3.8-A.) The existence of this narrow bottleneck is also visible in the $\mathbf{W}$ matrix (see Figure 3.11-A): a random walk initialized within the fat diagonal of the upper left corner of $\mathbf{W}$ (the low frequency part of the chirp) is trapped in this region of the

matrix, and can only travel along this fat diagonal. As a result, it will take many steps for the random walk to reach another slow patch $\mathbf{x}_m$ if $|n-m|$ is large. This notion can be quantified by computing the hitting time, $h(\mathbf{x}_n, \mathbf{x}_m)$, which measures the expected minimum number of steps that it takes for the random walk, started at vertex $\mathbf{x}_n$, to reach the vertex $\mathbf{x}_m$ [14]

$$h(\mathbf{x}_n, \mathbf{x}_m) = E_n \min\{j \geq 0 : Z_j = \mathbf{x}_m\},$$

where the expectation E_n is computed when the random walk is initialized at vertex $\mathbf{x}_n$, i.e. when $Z_0 = \mathbf{x}_n$. The commute time [14]: provides a symmetric version of h, and is defined by

$$\kappa(\mathbf{x}_n, \mathbf{x}_m) = h(\mathbf{x}_n, \mathbf{x}_m) + h(\mathbf{x}_m, \mathbf{x}_n). \tag{4.2}$$

4.2.2 Spectral representation of the commute time

When the Markov process is aperiodic and the graph is connected, the commute time can be expressed using the eigenvectors $\phi_1, \ldots, \phi_N$ of the symmetric matrix

$$\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} = \mathbf{D}^{1/2}\mathbf{P}\mathbf{D}^{-1/2}.$$

The corresponding eigenvalues can be labeled such that $-1 < \lambda_N \leq \ldots \leq \lambda_2 < \lambda_1 = 1$. Each eigenvector ϕ_k is a vector with N components, one for each vertex of the graph. Hence, we write

$$\phi_k = \begin{pmatrix} \phi_k(\mathbf{x}_1) & \phi_k(\mathbf{x}_2) & \ldots & \phi_k(\mathbf{x}_N) \end{pmatrix}^T,$$

to emphasize the fact that we consider ϕ_k to be a function sampled on the vertices of Γ. The commute time can be expressed as

$$\kappa(\mathbf{x}_n, \mathbf{x}_m) = \sum_{k=2}^{N} \frac{1}{1-\lambda_k} \left(\frac{\phi_k(\mathbf{x}_n)}{\sqrt{\pi_n}} - \frac{\phi_k(\mathbf{x}_m)}{\sqrt{\pi_m}} \right)^2, \tag{4.3}$$

where $\pi_n = \sum_{m=1}^{N} w_{n,m} / \sum_{j,l=1}^{N} w_{j,l}$ is the stationary distribution associated with the transition probability matrix in (4.1)$\mathbf{P}$ [55, 77].

4.2.3 The relationship to diffusion maps

The *diffusion distance* [23] between vertices $\mathbf{x}_m$ and $\mathbf{x}_n$, $D_t(\mathbf{x}_m, \mathbf{x}_n)$, measures the distance between the transition probability distributions – computed at time t – of two random walks initialized at $\mathbf{x}_n$ and $\mathbf{x}_m$, $\sum_{l=1}^{N} |\mathbf{P}_{n,l}^{(t)} - \mathbf{P}_{m,l}^{(t)}|^2$. The diffusion distance can also be decomposed in terms of the eigenvectors ϕ_k [23],

$$D_t^2(\mathbf{x}_m, \mathbf{x}_n) = \frac{1}{V} \sum_{k=2}^{N} \lambda_k^{2t} \left(\frac{\phi_k(\mathbf{x}_m)}{\sqrt{\pi_m}} - \frac{\phi_k(\mathbf{x}_n)}{\sqrt{\pi_n}} \right)^2, \tag{4.4}$$

where $V = \sum_{m',n'} w_{m',n'}$ is the volume of the graph. It follows that the commute time is a scaled sum of the squares of diffusion distances computed at all times,

$$\kappa(\mathbf{x}_m, \mathbf{x}_n) = V \sum_{t=0}^{\infty} D_{t/2}^2(\mathbf{x}_m, \mathbf{x}_n). \tag{4.5}$$

The significance of this equation is that the commute time includes the short term evolution ($t \approx 0$) as well as the asymptotic regime ($t \to \infty$) of the random walk. We will come back to this analysis in section 4.4.4.

4.3 Parametrizing the patch-graph

Equation (4.3) suggests the following embedding Ψ of the patch-graph Γ into $\mathbb{R}^{N-1}$,

$$\Psi : \mathbf{x}_n \longrightarrow \frac{1}{\sqrt{\pi_n}} \left(\frac{\phi_2(\mathbf{x}_n)}{\sqrt{1-\lambda_2}} \quad \frac{\phi_3(\mathbf{x}_n)}{\sqrt{1-\lambda_3}} \quad \dots \quad \frac{\phi_N(\mathbf{x}_n)}{\sqrt{1-\lambda_N}} \right)^T, \qquad n = 1, 2, \dots, N. \tag{4.6}$$

If we agree to measure the distance on the graph Γ using the square root of the commute time, then the mutual Euclidean distance after embedding is equal to the original distance on the graph,

$$\|\Psi(\mathbf{x}_n) - \Psi(\mathbf{x}_m)\| = \sqrt{\kappa(\mathbf{x}_n, \mathbf{x}_m)}. \tag{4.7}$$

The result is a direct consequence of (4.4) and (4.5). Similar ideas were first proposed in [11] to embed manifolds and are the foundation of the parametrizations given in [7, 23]. In practice, we need not use all the $N-1$ coordinates in the embedding defined by (4.6). Indeed, since $\lambda_N \leq \cdots \leq \lambda_2 < \lambda_1 = 1$, we have that $\frac{1}{\sqrt{1-\lambda_N}} \leq \cdots \leq \frac{1}{\sqrt{1-\lambda_3}} \leq \frac{1}{\sqrt{1-\lambda_2}}$, and therefore, if we can

accept some approximation error, then we can use only the first d' coordinates of Ψ. As we will see in section 4.4.4, this dimension reduction further improves the separation between slow patches and fast patches. In the remaining of the paper we will work with the embedding of Γ into $\mathbb{R}^{d'}$ defined by

$$\Phi : \mathbf{x}_n \longrightarrow \frac{1}{\sqrt{\pi_n}} \begin{pmatrix} \frac{\phi_2(\mathbf{x}_n)}{\sqrt{1-\lambda_2}} & \dots & \frac{\phi_{d'+1}(\mathbf{x}_n)}{\sqrt{1-\lambda_{d'+1}}} \end{pmatrix}^T . \tag{4.8}$$

We note that we can always choose d' such that the embedding Φ almost preserves the commute time,

$$\|\Phi(\mathbf{x}_n) - \Phi(\mathbf{x}_m)\|^2 \approx \kappa(\mathbf{x}_n, \mathbf{x}_m). \tag{4.9}$$

In fact, our experiments indicate that this approximation holds for small values of d'.

4.3.1 Examples (revisited)

Figure 4.1 displays the embedding of the patch-sets associated with signals and images A-F using the map Φ (4.8), where $d' = 3$. The blue curve in Figure 4.1-A corresponds to the slow patches (low frequencies of the chirp) that are connected according to their temporal proximity. On the other hand, red and orange patches extracted from the high frequency part of the chirp are now concentrated in a relatively small region (compare to Figure 3.8-A). Similar features are seen in the parametrizations of the patch-graphs associated with signals and images B-F.

4.4 A model for the patch-graph and the analysis of its embedding

4.4.1 Our approach

The embedding of the patch-graph Γ defined by Φ, in (4.8), should lead to a representation of the patch-set in $\mathbb{R}^{d'}$ where distances correspond to commute times measured on the graph before embedding. Our goal is to explain the concentration of the fast patches created by the embedding Φ (see e.g. Figure 4.1). Our approach is based on a theoretical analysis of a graph model that epitomizes the characteristic features observed in patch-graphs composed of a mixture of fast and slow patches. This model is composed of two subgraphs: a subgraph of *slow patches*, which are

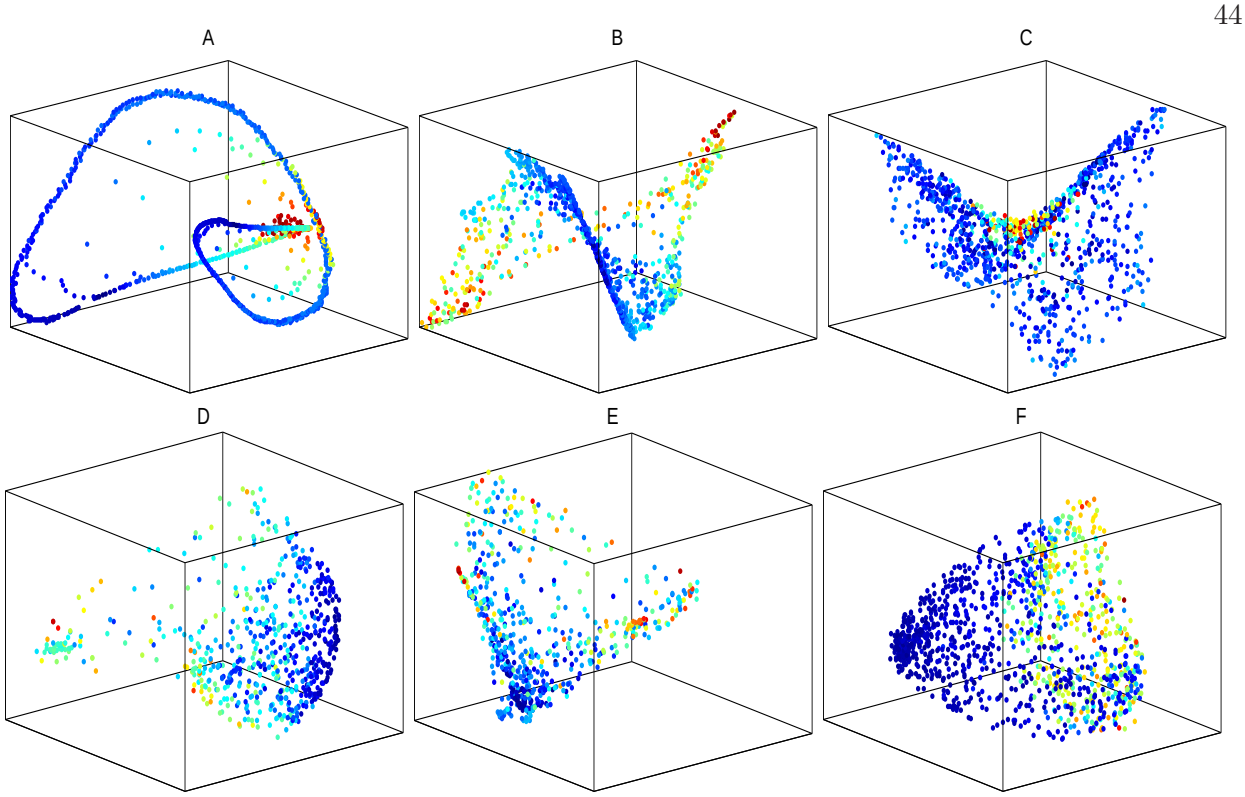

Figure 4.1: Scatter plot of the patch-set shown in Figure 3.8 after parametrizing using Φ in (4.8), with $d' = 3$. The fast patches (red and orange) are now concentrated and have been lumped together. The slow patches (blue-green) remain aligned along curves (for time-series) and surfaces (for images).

extracted from the smooth regions of the signal, and a subgraph of *fast patches*, which are extracted from the regions of the signal that contain singularities, changes in frequency, or energetic transients. We confirm our theoretical analysis with numerical experimentations using synthetic signals in section 4.5, and we demonstrate that our conclusions are in fact applicable to a larger class of patch-graphs. The graph models are introduced in section 4.4.2. Our theoretical analysis of the embedding of the graph models is given in section 4.4.3. We evaluate the performance of the embedding Φ when d' is small in section 4.4.4.

4.4.2 The prototypical graph models

We define the graph models in terms of the nonzero entries in the associated weight matrix $\mathbf{W}$. Without loss of generality, we assume that the number of vertices N is even.

The slow graph model. The large entries in a weight matrix $\mathbf{W}$ of a patch-graph composed only of slow patches will have large entries when $|n-m|$ is small[1] : temporal/spatial proximity implies proximity in patch-space (see e.g. Figure 3.11-A, top corner). We therefore define the *slow graph model* as follows.

Definition 6. *The slow graph $\mathcal{S}(N, L)$ is a weighted graph composed of N vertices $\mathbf{x}_1, \ldots, \mathbf{x}_N$. The weight on the edge $\{\mathbf{x}_n, \mathbf{x}_m\}$ is defined by*

$$w_{n,m} = \begin{cases} w_{\mathcal{S}} & \text{if } |n-m| \leq L, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{for} \quad 1 \leq n, m \leq N \quad \text{and} \quad 2L+1 \leq N. \tag{4.10}$$

The weight w_s is a positive real number that models the edge weight between two temporally adjacent patches. The parameter L characterizes the thickness of the diagonal in $\mathbf{W}$. The slow graph is connected and each vertex has at most $2L$ neighbors, not including self-connections (see Figure 4.2). Hence, we require that $2L+1 \leq N$. Finally, note that the slow graph is distinct from a regular ring, since the first and last vertices are not connected. We do not consider a regular ring since it would imply that the underlying signal is periodic.

The fast graph model. We now consider the model for a patch-graph built from a patch-set comprising only fast patches. As demonstrated in section 3.7, most of the entries in $\mathbf{W}$ have similar sizes, and appear to be scattered throughout the matrix: temporal/spatial proximity does not correlate with proximity in $\mathbb{R}^d$. In fact, fast patches are all far away from one another. We therefore define the *fast graph model* as follows.

Definition 7. *The fast graph $\mathcal{F}(N, p)$ is a random weighted graph composed of N vertices, $\mathbf{x}_1, \ldots, \mathbf{x}_N$.*

[1] We assume that the rows/columns of $\mathbf{W}$ are ordered according to increasing index n of the sequence $\{x_n\}$. This assumption does not affect the graph's parametrization nor our theoretical conclusions, but allows us to interpret the structure in $\mathbf{W}$.

The weight on the edge $\{\mathbf{x}_n, \mathbf{x}_m\}$ *is defined by*

$$w_{n,m} = w_{m,n} = \begin{cases} w_{\mathcal{F}} & \textit{with probability } p, \\ 0 & \textit{with probability } 1-p \end{cases} \qquad \textit{if } 1 \le n < m \le N,$$

and

$$w_{n,m} = 1 \quad \textit{if} \quad n = m.$$

The weight $w_{\mathcal{F}}$ is a positive real number that models the distance between two fast patches. The fast graph model is equivalent to a weighted version of the Erdös-Renyi graph model [32], except that $\mathcal{F}(N,p)$ contains self-connections. The parameter p controls the density of the edges; $p = 1$ corresponds to a connected graph (clique).

The fused graph model. The *fused graph model* exemplifies the patch-set associated with a signal, or an image, which exhibits regions of fast and slow changes. The fused graph combines a slow and a fast subgraph of equal size (see Figure 4.2).

Definition 8. *The fused graph* $\Gamma^*(N)$ *is a weighted graph composed of a slow subgraph* $\mathcal{S}(N/2, L)$ *and a fast subgraph* $\mathcal{F}(N/2, p)$*. In addition, edges between* $\mathcal{S}(N/2, L)$ *and* $\mathcal{F}(N/2, p)$ *are created randomly and independently with probability* q *and assigned the edge weight* $w_c > 0$*.*

Edges between $\mathcal{S}(N/2, L)$ and $\mathcal{F}(N/2, p)$ ensure a high probability that $\Gamma^*(N)$ is connected (a requirement for the validity of the parametrization (4.6)). These edges allow us to model patches that are extracted from regions of the image that combine edges/transients and smooth intensity. If no edges are created between the two subgraphs because q is too small, then an edge is placed at random between the two subgraphs to ensure that the final fused graph is connected.

The true patch-graph is always constructed using a ν nearest neighbor rule (see section 2.2): each patch is connected to at least ν other patches. In order to mimic a true patch-graph, we adjust the thickness L of the slow subgraph to the density of the edge connection, p, in the fast subgraph, so that on average, each vertex in the fused graph is connected to $2L$ vertices. We know that the number of edges between distinct vertices in $\mathcal{F}(N,p)$ is a binomial random variable with expectation

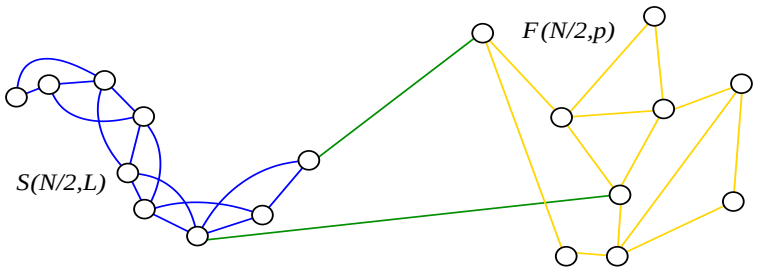

Figure 4.2: The fused graph model $\Gamma^*(N)$ is composed of a slow graph $\mathcal{S}(N/2, L)$ (blue) and a fast graph $\mathcal{F}(N/2, p)$ (orange), connected by random edges (green).

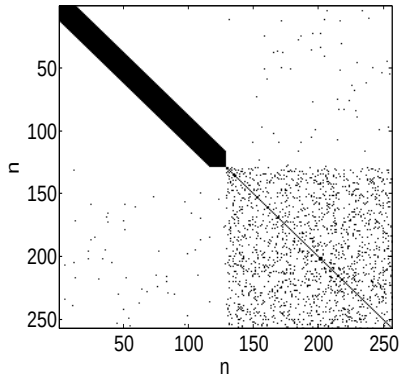

Figure 4.3: The weight matrix $\mathbf{W}$ of the fused graph model $\Gamma^*(256)$ is displayed as an image: $w_{n,m}$ is encoded as a grayscale value: from white ($w_{n,m} = 0$) to black ($w_{n,m} = 1$). The entries of $\mathbf{W}$ associated with the slow graph appear in the upper-left quadrant of $\mathbf{W}$. Entries associated with the fast graph appear in the lower right quadrant. Random edges between the fast graph and slow graph appear in the upper right and lower left quadrants.

$\frac{N(N-1)}{2}p$. Since the total number of edges between distinct vertices of $\mathcal{S}(N, L)$ is equal to[2]

$$\sum_{j=1}^{L}(N-j) = NL - \frac{L(L+1)}{2}, \tag{4.11}$$

we choose

$$p = \frac{2L}{N-1} - \frac{L(L+1)}{N(N-1)}. \tag{4.12}$$

This choice of p guarantees that the total expected number of edges in $\mathcal{F}(N, p)$ is equal to the total number of edges in $\mathcal{S}(N, L)$. Furthermore, provided that $L = \mathcal{O}(\ln(N))$, a short computation shows

[2] This is equivalent to the number of entries along the first L upper diagonals of the matrix $\mathbf{W}$.

that, for large values of N, this choice of p also ensures that the expected degree of a vertex in $\mathcal{F}(N,p)$ is equal to the average degree of a vertex in $\mathcal{S}(N,L)$. Figure 4.3 shows the nonzero entries in the weight matrix associated with one realization of the fused graph model using parameters $N = 256$, $L = \lceil 2 \ln N \rceil = 12$ and $q = \frac{1}{N}$. Vertices $\mathbf{x}_n$ with $n \leq 128$ are only connected to other vertices $\mathbf{x}_m$ if $|n - m| \leq L$. This connectivity mimics the temporal connectivity present in the smooth parts of a signal or image (compare with Figure 3.11).

4.4.3 The main result

Our goal is to understand the effect of the embedding Φ defined by (4.8) on the fused graph. It turns out that studying the embedding of each individual subgraph (slow and fast) separately is much more tractable than considering the entire fused graph. To complement our theoretical study of the fast and the slow subgraphs, we provide numerical evidence in sections 4.4.4 that indicates that our understanding of the embedding of the subgraphs can be used to analyze the embedding of the fused graph. In section 4.5, we confirm that our theoretical analysis can be applied to true patch-graphs. Instead of studying Φ directly, we take advantage of the fact that the embedding Φ almost preserves the commute time (see (4.9)). We can therefore understand the effect of the embedding on the distribution of mutual distances $\|\Phi(\mathbf{x}_n) - \Phi(\mathbf{x}_m)\|$ within a subgraph by studying the distribution of the commute times $\kappa(\mathbf{x}_n, \mathbf{x}_m)$ on that subgraph. While it would appear that it is a straightforward affair to compute the commute time on the slow graph, the computation becomes rapidly intractable. For this reason we provide lower and upper bounds for the average commute time on the slow and fast subgraphs, respectively. This is sufficient for our needs, since the two bounds rapidly separate even for low values of N. To estimate these bounds, we rely on the connection between commute times on a graph and effective resistance on the corresponding electrical network [17, 29]. Specifically, we map a graph to an electrical circuit as follows: each edge with weight $w_{n,m}$ becomes a resistor with resistance $1/w_{n,m}$. The vertices of the graph are the connections in the circuit. Given two vertices, $\mathbf{x}_n$ and $\mathbf{x}_m$ that are identified as terminals in the circuit, one can compute the effective resistance between these terminals, $R_{n,m}$. The key result

is that $\kappa(\mathbf{x}_n, \mathbf{x}_m) = VR_{n,m}$, where V is the volume of the graph [17].

Before stating the main Lemma, let us take a moment to compute some rough estimates of the commute times on the slow and fast graphs. To get some quick answers, we consider the simplest versions of the two graph models. When $L = 1$, the slow graph $\mathcal{S}(N, 1)$ is a *path* with self-connections. On a path of N vertices *without* self-connections, the commute time between vertex $\mathbf{x}_n$ and $\mathbf{x}_m$ is equal to $2(N-1)|m-n|$ [55]. Therefore, the average commute time (computed over all pairs of vertices) on a path of length N is $\mathcal{O}(N^2)$. While it would make sense that adding edges to a path should decrease the commute time, this is usually not true [55]. Nevertheless, the presence of edges that allow the random walk to move forward by a distance L at each time step lead us to conjecture that the average commute time on $\mathcal{S}(N, L)$ should at least decrease by a factor of L. In fact, as we will see in Lemma 2, the average commute time of the slow graph decreases by a factor of L^2. With regard to the fast graph, we can analyze the case where the density of edges $p = 1$. In this case, the fast graph $\mathcal{F}(N, 1)$ is a *complete graph*, or *clique*, and every vertex is connected to every other vertex. In a complete graph, the average commute time is $\mathcal{O}(N)$ [55]. Since the fast graph can be regarded as a complete graph whose edges have been removed with probability $1 - p$, we expect the commute time to be slightly larger than $\mathcal{O}(N)$, because removing edges restricts the random walker's options to get from one vertex to another. Again, in agreement with our intuition, Lemma 2 asserts that in the fast graph, the commute time increases by a factor of roughly $[L \ln(N)/\ln(L)]$.[3]

We are now ready to state the main lemma. Our results will be stated in terms of the "average behavior" of the commute time on each graph, a concept that we need to define properly. In the case of the slow graph, which is deterministic, we consider the average commute time computed over all pairs of vertices.

Definition 9. *Let $\kappa_{\mathcal{S}}$ be the average commute time between vertices in the slow graph $\mathcal{S}(N, L)$*

$$\kappa_{\mathcal{S}} \triangleq \frac{2}{N(N-1)} \sum_{1 \le m < n \le N} \kappa(\mathbf{x}_n, \mathbf{x}_m). \tag{4.13}$$

[3] Since $p = p(L)$ as defined in (4.12), the increase of the commute time in $\mathcal{F}(N, p)$ also depends on L.

In the case of the fast graph, the "average behavior" of the commute time needs to be defined more carefully. Indeed, each fast graph is a realization of a stochastic process, and therefore we need to consider the *expectation* of the commute time. More precisely, given a realization, $\mathcal{F}$, of a fast graph, we compute the expected commute time $\mathrm{E}_{\mathbf{x}_n,\mathbf{x}_m}[\kappa|\mathcal{F}]$ as the expectation of $\kappa(\mathbf{x}_m,\mathbf{x}_n)$ over all possible random assignment of the vertices $\mathbf{x}_n$ and $\mathbf{x}_m$. We then need to consider how $\mathrm{E}_{\mathbf{x}_n,\mathbf{x}_m}[\kappa|\mathcal{F}]$ varies as a function of $\mathcal{F}$. Therefore, we compute a second expectation over all possible fast graphs $\mathcal{F}$.

Definition 10. *The expected commute time $\kappa_{\mathcal{F}}$ on a fast graph $\mathcal{F}$ generated according to Definition 7 is defined by*

$$\kappa_{\mathcal{F}} \triangleq E_{\mathcal{F}}\left[E_{\mathbf{x}_n,\mathbf{x}_m}[\kappa|\mathcal{F}]\right], \tag{4.14}$$

where the inner expectation is computed over all random assignments of the vertices $\mathbf{x}_n, \mathbf{x}_m$ given a realization $\mathcal{F}$ of a fast graph geometry, and the outer expectation is computed over all possible realizations $\mathcal{F}$ of the fast graph.

Lemma 2. *We have*

$$\left(N(2L+1)-L(L+1)\right)\frac{2\left(N+1\right)}{3L^2(L+1)} \leq \kappa_{\mathcal{S}}. \tag{4.15}$$

We also have

$$\kappa_{\mathcal{F}} \leq \left(N(2L+1)-L(L+1)\right)\left(\frac{\ln N}{\ln\left(2L-\frac{L(L+1)}{N}+1\right)}+\frac{1}{2}\right), \tag{4.16}$$

provided that, for all assignments of the vertices $\mathbf{x}_m$ and $\mathbf{x}_n$, and for all fast graphs $\mathcal{F}$, the covariance $\mathrm{Cov}(M, R_{m,n})$ between the number of edges, M, and the effective resistance, $R_{m,n}$, of the associated electrical circuit is nonpositive.

Proof: *See appendix A.6.*

It is clear that if L is held constant while N increases, then p will approach zero, according to (4.12). If p approaches zero, then the fast graph is more likely to be disconnected, which is contrary to our assumptions on a general patch-graph. To avoid this, L must change with N. As shown in appendix A.5), choosing $L > \ln N$ is sufficient to guarantee a vanishing probability of the fast

graph being disconnected. It can be verified that if $L = N^{1/3}$, then the upper bound on $\kappa_{\mathcal{F}}$ grows at the same rate as the lower bound on $\kappa_{\mathcal{S}}$. Therefore, to ensure that the upper bound on $\kappa_{\mathcal{F}}$ is negligible relative to the lower bound on $\kappa_{\mathcal{S}}$, when N is large, while still maintaining a connected fast graph with high probability, we must choose $\ln N < L < N^{1/3}$. Indeed, we will fix $L = c \ln N$ for some constant $c > 1$.

Corollary 1. *Assume that $L = c \ln N$ for some constant $c > 1$. It follows that, as $N \to \infty$, the lower bound on $\kappa_{\mathcal{S}}$ grows like $\left(\frac{N}{\ln N}\right)^2$, and the upper bound on $\kappa_{\mathcal{F}}$ grows like $\frac{N(\ln N)^2}{\ln \ln N}$. Furthermore, the lower bound on $\kappa_{\mathcal{S}}$ grows faster than the upper bound on $\kappa_{\mathcal{F}}$, and so with a probability that approaches one as $N \to \infty$,*

$$\frac{\kappa_{\mathcal{F}}}{\kappa_{\mathcal{S}}} \to 0.$$

Proof: *Notice that $\kappa_{\mathcal{S}}$ is bounded away from zero. Because the choice of L guarantees that the fast graph is connected with a probability approaching one, $\kappa_{\mathcal{F}}$ is finite with probability approaching one. Therefore the ratio $\kappa_{\mathcal{F}}/\kappa_{\mathcal{S}}$ is bounded below by zero and from above by a ratio of the bounds from Lemma 2. The ratio of bounds goes to zero, which follows from a simple, but lengthy, limit calculation.* □

We can translate the corollary in terms of the mutual distances between vertices of the subgraphs after the embedding Φ: $\Phi(\mathcal{F}(N,p))$ will be more concentrated than $\Phi(\mathcal{S}(N,L))$.

4.4.4 Spectral decomposition of commute times on the graph models

The results of section 4.4.3 apply to the exact commute times on the graph models. However, as mentioned in section 4.3, it is more practical to use a truncated version of the spectral expansion of the commute time, defined by Equation (4.3). We also noticed that the commute time encompasses the short term evolution ($t \approx 0$) as well as the asymptotic regime ($t \to \infty$) of the behavior of the random walk. Neglecting eigenvalues ϕ_k for large k emphasizes the long term behavior of the random walk, and we expect that it should further increase the difference between the dynamics of the random walk on the slow and fast graphs. In this section, we confirm experimentally

that approximating the commute times by truncating the expansion (4.3) actually emphasizes the separation between average commute times on the fast subgraph and the slow subgraph in the fused graph model. In all the numerical experiments in this section, unless otherwise stated, we fix $N = 1024$, $L = \lceil 2\ln(N) \rceil$, p is chosen according to (4.12), $q = 1/N$, and $w_{\mathcal{S}} = w_{\mathcal{F}} = w_c = 1$. In all experiments, we compute the eigenvalues $\{\lambda_k\}$ of the matrix $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ associated with the fast graph, the slow graph, and the fused graph.

4.4.4.1 Slow and fast subgraphs: two different dynamics revealed by the spectral decomposition

We first provide a back-of-the-envelope computation of the spectrum of the slow and fast graphs. As we have noticed before, the slow graph model is a "fat" path. We know that the spectrum of a path without self-connections [20] is given by

$$\cos\left[\pi(k-1)/(N-1)\right], \quad k = 1, 2, \ldots, N.$$

We expect therefore that the eigenvalues associated with the slow graph will decay slowly away from one for small k. Figure 4.4 (inset) displays the eigenvalues associated with the slow graph model. As expected, the spectrum is flat around $k = 0$ and exhibits the slowest decay of all the graph models. We use the similarity between the fast graph model and the Erdös-Renyi graph to predict the spectrum of the fast graph. Except for $\lambda_1 = 1$, all the other eigenvalues of an Erdös-Renyi graph asymptotically follow Wigner's semicircle distribution [21]. Our numerical experiments confirm this prediction: as shown in Figure 4.4-right, the eigenvalues of the fast graph appear to be distributed along a semicircle.

The decay of the spectrum has a direct influence on the dynamics of the random walk. Specifically, the spectral gap controls the *mixing rate*, which measures the expected number of time-steps that are necessary to reduce the distance between the probability distribution after t steps $\mathbf{P}^{(t)}_{n,m}$ and the stationary distribution π_m by a certain factor [91]. This concept is justified by

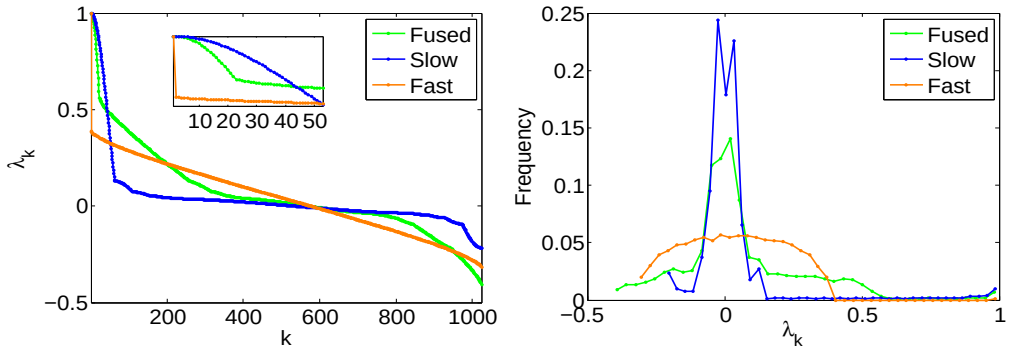

Figure 4.4: The eigenvalues λ_k of the matrix $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ associated with the fused (green), slow (blue), and fast (orange) graphs. Left: λ_k as a function of k; right: histogram of the λ_k.

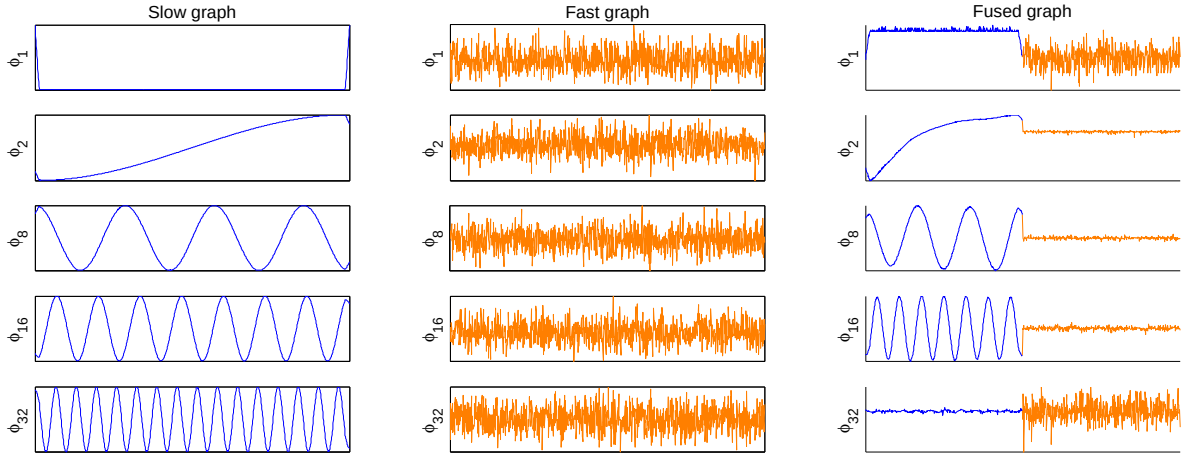

Figure 4.5: The eigenvectors $\{\phi_1, \phi_2, \phi_8, \phi_{16}, \phi_{32}\}$ associated with the slow (left), fast (center), and fused (right) graphs. Right: the large amplitude of the eigenvectors ϕ_k on the first half of vertices (blue) belonging to the slow subgraph leads to a larger separation between the fast and slow subgraphs when truncating the commute time expansion.

the fact that the convergence of $\mathbf{P}^{(t)}_{n,m}$ is exponential [30], and is given by

$$\max_{n,m} \left| \frac{\mathbf{P}^{(t)}_{n,m}}{\pi_m} - 1 \right| \leq \frac{\lambda^t_{max}}{\pi_{min}}, \quad t = 1, 2, \ldots \tag{4.17}$$

where $\lambda_{max} = \max\{\lambda_2, |\lambda_N|\}$ (which is related to the spectral gap), and π_{min} is the smallest entry of the stationary distribution. Since λ_2 is much larger in the slow graph than in the fast graph,

we expect that convergence to the associated stationary distribution will take longer on the slow graph than on the fast graph.

4.4.4.2 The dynamics of the fused graph is enslaved by the slow subgraph

We now consider a random walk on the fused graph. If this random walk begins at $\mathbf{x}_n$ in the fast subgraph of the fused graph, then after a small number of steps, t_0, the probability of finding the random walker at any other vertex $\mathbf{x}_m$ in the fast subgraph is close to the stationary distribution, $\mathbf{P}^{t_0}_{n,m} \approx \pi_m$. On the other hand, during the same amount of steps, a random walk initialized in the slow subgraph will only explore a small section of the slow subgraph, and consequently, the transition probabilities will still be similar to its initial values $\mathbf{P}^{(t_0)}_{n,m} \approx \mathbf{P}_{n,m}$. As a result, the restriction imposed by the geometry of the slow subgraph is expected to decrease the convergence rate of the transition probabilities on the fused graph. We confirm this analysis with experimental results. Figure 4.4 (inset) shows that for $k < 23$ the eigenvalues associated with the fused graph and the eigenvalues associated with the slow graph exhibit slow decay away from one, thereby increasing the convergence rate given in (4.17). For $25 \leq k \leq 400$, the eigenvalues of the fused graph decay at a rate similar to that of the fast graph. Finally, for $k \geq 400$ the eigenvalues of the fused graph join those of the slow graph (see also the histogram in Figure 4.4-right). We have observed in numerical experiments that these transitions in the behavior of the spectrum of the fused graph are not affected by varying the parameters N, L, and q.[4] We conclude that the slow subgraph has the largest influence on the first few (small k) eigenvalues λ_k of the fused graph.

4.4.4.3 The eigenvectors of the fused graph and their impact on the commute time

The transition exhibited in the spectrum of the fused graph can also be detected in the corresponding eigenvectors ϕ_k. Figure 4.5 shows the eigenvectors $\{\phi_1, \phi_2, \phi_8, \phi_{16}, \phi_{32}\}$ corresponding to the three graph models. The first eigenvector ϕ_1 has entries equal to the square root of the stationary distribution, $\phi_1(\mathbf{x}_n) = \sqrt{\pi_n}$, and is not used in the expansion of the commute time (4.3).

[4] We assume q is small because having a large q would increase the expected degree of a vertex in the fused graph, which is contrary to a true patch-graph, in which each vertex has roughly ν neighbors.

As expected, the random walk spends most of its time inside the slow subgraph of the fused graph, as indicated by the larger values of ϕ_1 for the first (blue) $N/2$ vertices (see Figure 4.5-right). The eigenvectors $\{\phi_2, \phi_8, \phi_{16}\}$ of the fused graph exhibit large amplitude oscillations over the vertices belonging to the slow subgraph (first half – shown in blue – of the plots in Figure 4.5-right), which resemble those found in the eigenvectors associated with the slow graph (Figure 4.5-left). As k increases, the eigenvectors ϕ_k of the fused graph become more and more similar to the eigenvectors of the fast graph.

The impact of the eigenvectors ϕ_k on the commute time on the fused graph can be analyzed by estimating the size of the terms

$$\frac{1}{1-\lambda_k}\left(\frac{\phi_k(\mathbf{x}_n)}{\sqrt{\pi_n}} - \frac{\phi_k(\mathbf{x}_m)}{\sqrt{\pi_m}}\right)^2 \tag{4.18}$$

in the spectral expansion (4.3) of the commute time κ. We claim that $\kappa(\mathbf{x}_n, \mathbf{x}_m)$ will be small if both vertices $\mathbf{x}_n$ and $\mathbf{x}_m$ are in the fast subgraph, and that κ will be large if either vertex is in the slow subgraph.

We can first estimate the size of $\phi_k(\mathbf{x}_n)/\sqrt{\pi_n} - \phi_k(\mathbf{x}_m)/\sqrt{\pi_m}$. We observe that the eigenvectors ϕ_k for small values of k have large amplitude oscillations on vertices belonging to the slow subgraph, but are relatively constant on the fast subgraph (see Figure 4.5-right). Therefore, for small values of k, each term (4.18) will be small when $\mathbf{x}_n$ and $\mathbf{x}_m$ both belong to the fast subgraph (we also have $\pi_n \approx \pi_m$ when two vertices belong to the same subgraph). Conversely, these terms will be large when either $\mathbf{x}_n$ or $\mathbf{x}_m$ belongs to the slow subgraph. While this analysis of the size of the terms (4.18) only holds for small values of k, it turns out that these are the terms that have the largest influence in the expansion of the commute time (4.3). Indeed, the spectrum of the fused graph decays slowly, and therefore the first few coefficients $(1-\lambda_k)^{-1}$ in the commute time expansion (4.3) are much larger than the remainders, and therefore the terms (4.18) for small values of k will provide the largest contribution in the expansion of the commute time.

We conclude that $\kappa(\mathbf{x}_n, \mathbf{x}_m)$ is small when $\mathbf{x}_n$ and $\mathbf{x}_m$ belong to the fast subgraph, and $\kappa(\mathbf{x}_n, \mathbf{x}_m)$ is large when either vertex is in the slow subgraph. Furthermore, we expect that this

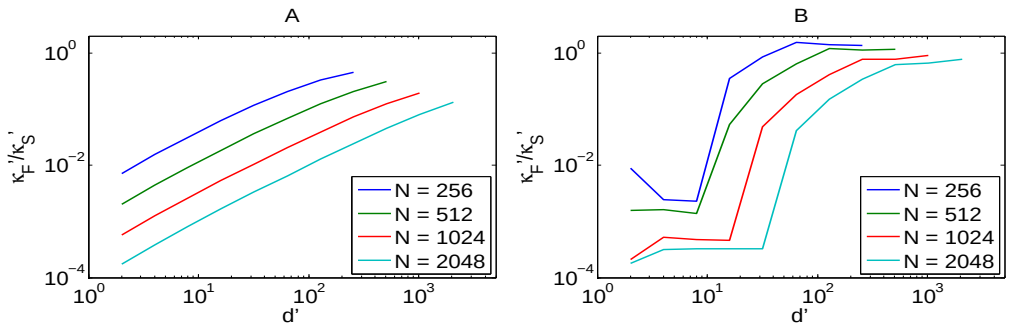

Figure 4.6: $\kappa'_{\mathcal{F}}/\kappa'_{\mathcal{S}}$ as a function of the dimension d' of the embedding Φ, for several values of the number of vertices N. Left: slow $\mathcal{S}$ and fast $\mathcal{F}$ graphs separately; right: slow and fast subgraphs in the fused graph Γ^*.

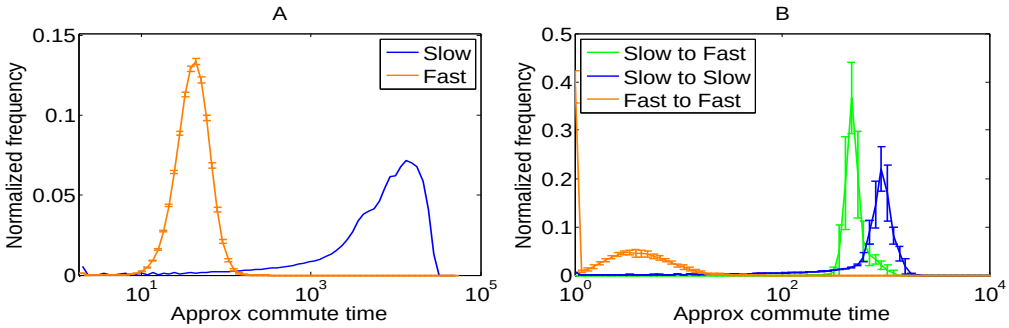

Figure 4.7: Histogram of κ'. Left: slow graph $\mathcal{S}$ and fast graph $\mathcal{F}$. Right: κ' for the three types of transition between the subgraphs of the fused graph Γ^*. Error bars represent one sample standard deviation using 25 realizations. Note the logarithmic scale on the horizontal axes.

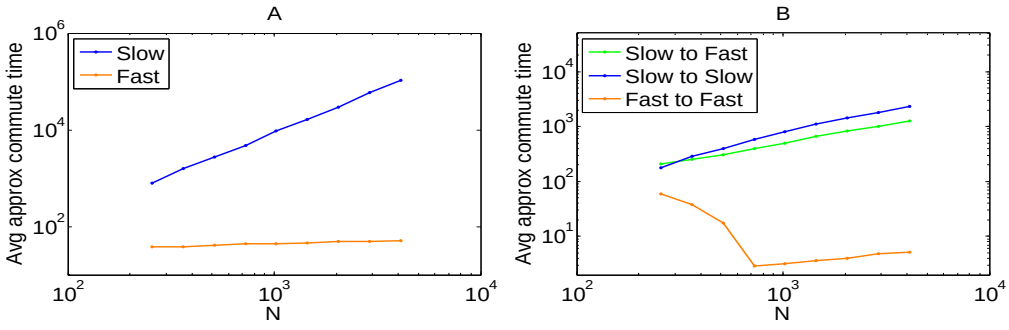

Figure 4.8: κ' as function of N. Left: slow graph $\mathcal{S}$ and fast graph $\mathcal{F}$. Right: κ' for the three types of transition between the subgraphs of the fused graph Γ^*.

difference will be further magnified if we replace the exact expansion of κ in (4.3) by an approximation that only includes the first few values of k.

4.4.4.4 The truncated spectral expansion of the commute time increases the contrast between the slow and fast subgraphs

We finally come to the heart of the section: the numerical computation of the average approximate commute time defined by

$$\kappa' = \frac{2}{N(N-1)} \sum_{n<m} \|\Phi(\mathbf{x}_n) - \Phi(\mathbf{x}_m)\|^2. \tag{4.19}$$

Because of (4.7), we expect that κ' will be close to the true commute time κ. We compute κ' for the three graph models: slow, fast and fused. We generated 25 realizations of the fast and fused graphs, and we estimated the expected commute time with the sample mean, given by κ' in (4.19).

Figure 4.6-A displays $\kappa'_{\mathcal{F}}/\kappa'_{\mathcal{S}}$ as a function of the number of terms d' used in the embedding (4.8), for several values of the number of vertices N, for the slow and fast graphs. Our theoretical analysis of $\kappa_{\mathcal{F}}/\kappa_{\mathcal{S}}$, performed in Corollary 1, is only valid for large values of N. Nevertheless, our numerical simulations indicate that for very low values of N, $\kappa_{\mathcal{F}}$ is already smaller than $\kappa_{\mathcal{S}}$, since all ratios are below one (see Figure 4.6-A). Furthermore, we see that this ratio is even smaller for smaller values of d'. We observe similar results when the commute times $\kappa'_{\mathcal{S}}$ and $\kappa'_{\mathcal{F}}$ are computed within the slow the fast subgraphs of the fused graph (see Figure 4.6-B). These results confirm that the embedding Φ will further concentrate the vertices of the fast graph if d' is chosen to be much smaller than N. We have observed experimentally that choosing $d' \approx \ln(N)$ leads to the smallest ratio of averages, not only on the graph models, but also on the general patch-graphs studied in section 4.5.

The enslaving of the fused graph by the slow graph is clearly shown in Figure 4.7-B, where the normalized histogram of κ' is shown for the three types of transition between the subgraphs of the fused graph Γ^*: slow $\rightarrow$ slow, fast $\rightarrow$ fast, and slow $\rightarrow$ fast. The histogram of the slow $\rightarrow$ fast transition is very similar to the histogram of the slow $\rightarrow$ slow transition, clearly indicating that

once the random walk is trapped in the slow subgraph, the presence of the fast subgraph does not help the random walk escape from the slow graph. We also notice that the average of κ' for the fast $\rightarrow$ fast transition is roughly two orders of magnitude smaller than the average of κ' for the slow $\rightarrow$ slow, or slow $\rightarrow$ fast transitions. In addition, the variance of each distribution is small enough to limit the overlap between the distributions.

Figure 4.8 displays κ' as a function of the number of vertices N, where $d' = \ln N$. Again, this result confirms that the asymptotic analysis of the ratio $\kappa_{\mathcal{F}}/\kappa_{\mathcal{S}}$, performed in Corollary 1, actually holds for very small values of N. Indeed, whether the slow and fast graphs are considered separately (Figure 4.8-A), or are the components of the fused graph (Figure 4.8-B), the ratio $\kappa_{\mathcal{F}}/\kappa_{\mathcal{S}} \rightarrow 0$ (note the logarithmic scale). It is important to bear in mind that when analyzing images, N is typically of the order of 10^6 and therefore our theoretical analysis will hold without any difficulty. Lastly, we again note in Figure 4.8-B that the transitions slow $\rightarrow$ fast in the fused graph have the same dynamics as the transition slow $\rightarrow$ slow.

4.5 Numerical experiments with synthetic signals

In this section we validate our theoretical results using synthetic signals. Each signal is the realization of a stochastic process with a prescribed autocorrelation function. We study two types of stochastic processes: one that generates signals that transition from low to high local frequency, and a second one that yields signals with varying local smoothness. We argue that these signals embody the types of local changes that are of fundamental importance in many areas of image processing. For both classes of signals, we embed the patch-sets using Φ in (4.8). We study the property of the embedding by quantifying the average commute time κ', defined in (4.19) between fast and slow patches, and we compare the numerical results with the theoretical predictions given in section 4.4.

4.5.1 The signals

We consider two types of models: a time-frequency signal model and a local regularity signal model. Each model is characterized by an autocorrelation function. The autocorrelation function can be modified using a covariance parameter that controls the local frequency, or the local regularity of the signal. We partition the interval $[0, 1]$ into subintervals over which the covariance parameter is kept constant. The covariance parameter alternates between two different values creating subintervals of alternating local frequency, or alternating local regularity. The number of alternations is chosen randomly according to a homogeneous Poisson process with intensity μ: there are on the average $\mu+1$ subintervals. A simpler version of this model has been used in [22] to mimic the presence of edges in images. Unlike the model used in [22], we adjust the signal defined on each subinterval so that the result is continuous on $[0, 1]$. In all experiments that we report here we use $\mu = 3$. The autocorrelation function associated with the time-frequency signal model is given by

$$\mathrm{E}(x(t)\overline{x(t+\tau)} = 2\left(\frac{1+\cos(2\pi\tau)}{2}\right)^{\beta} - 1, \tag{4.20}$$

where $\tau \in [0, 1)$, and $\beta \geq 0$. As the covariance parameter β increases, the range of frequencies present in the signal also increases. Figure 4.9 displays a realization of this model where the signal's covariance parameter in (4.20) alternates four times between $\beta_{\mathcal{S}} = 8$, and $\beta_{\mathcal{F}} = 256$. See appendix A.7 for more on generating a signal from the time-frequency signal model. The autocorrelation function associated with the local regularity signal model is equal to that of fractional Brownian motion, given by

$$\mathrm{E}(x(\tau_1)\overline{x(\tau_2)}) = \frac{1}{2}\big(|\tau_1|^{2H} + |\tau_2|^{2H} - |\tau_2 - \tau_1|^{2H}\big), \tag{4.21}$$

where H is the Hurst parameter. As H decreases, the local regularity decreases. A realization of this model is shown in Figure 4.10 where the signal's covariance parameter alternates four times between $H_{\mathcal{S}} = 0.9$ and $H_{\mathcal{F}} = 0.3$. We use the method described in [2] to generate the fractional Brownian motion.

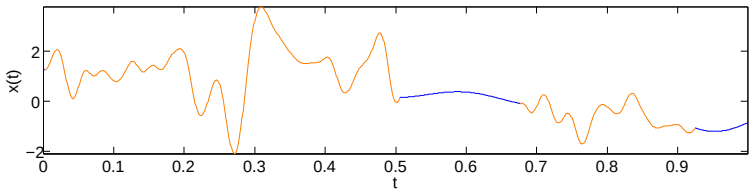

Figure 4.9: A realization of the time-frequency model. The low frequency portion ($\beta_{\mathcal{S}} = 8$) is shown in blue; the high frequency portion ($\beta_{\mathcal{F}} = 256$) is shown in orange. There are four subintervals ($\mu = 3$).

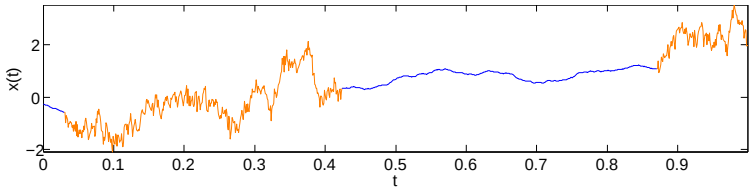

Figure 4.10: A realization of the local regularity model. The smooth portion ($H_{\mathcal{S}} = 0.9$) is shown in blue; the irregular portion ($H_{\mathcal{F}} = 0.3$) is shown in orange. There are four subintervals ($\mu = 3$).

4.5.2 Embedding the patch-graph

For each realization of a specific signal model, we construct a patch-set of $N = 1024$ maximally overlapping patches. The patch size is given by $d = 32$ for the time-frequency model, and $d = 16$ for the local regularity model. We compute the embedding Φ (4.8) and keep d' eigenvectors ϕ_k. Figure 4.11 shows the patch-set associated with the realization of the time-frequency signal displayed in Figure 4.9 before (left) and after (right) embedding. The scatterplot before embedding is computed using the first three principal components. Figure 4.12 shows the patch-set associated with the realization of the local regularity signal displayed in Figure 4.10 before (left) and after (right) embedding. The fast patches of the time-frequency signal are the orange patches extracted from the high frequency segments. The slow patches are the blue patches extracted from the low frequency sections. Similarly, the fast patches of the local regularity signal are the orange patches extracted from the irregular segments, and the slow patches are the blue patches extracted from the smooth

sections. For both signals, the fast patches are scattered across the space before embedding. After embedding, the fast patches are tightly clustered. This visual impression is confirmed by computing the mutual distance between patches after embedding, $\|\Phi(\mathbf{x}_n) - \Phi(\mathbf{x}_m)\|$. In principle, we should report the value of the Lipschitz ratio $\|\Phi(\mathbf{x}_n) - \Phi(\mathbf{x}_m)\|/\|\mathbf{x}_n - \mathbf{x}_m\|$ to quantify the contraction experienced through the mapping Φ. However, we have noticed that because the mutual distances $\|\mathbf{x}_n - \mathbf{x}_m\|$ between fast patches is typically large (as explained in section 3.7), the Lipschitz ratio ends up being small for fast patches. Therefore studying the size of the Lipschitz ratio associated with Φ does not reveal whether the map concentrates the fast patches or not, but only indicates that the sampling of the fast patches (in the patch-set) is coarse. For this reason we prefer to study how $\|\Phi(\mathbf{x}_n) - \Phi(\mathbf{x}_m)\|$ varies for pairs of slow and fast patches. Based on our theoretical analysis, we expect that after the embedding, the mutual distance between fast patches will becomes much shorter than the mutual distance between slow patches.

We point out that the eigenvectors ϕ_k used in the embedding Φ (4.8) are designed to have, on average, small gradients (as measured along edges of the graph). Indeed, these eigenvectors are also the eigenvectors of the graph Laplacian [20], and therefore minimize a Rayleigh ratio that quantifies the average norm of the gradient of ϕ_k. Thus, if we further restricted our computation of the commute times inside each subset of fast and slow patches to only those patches that were connected by an edge in the graph, we would expect to see smaller values and little dependence on whether or not the patch was fast or slow. However, since our theoretical analysis of section 4.4 is based on the average commute time between all vertices belonging to the fast or slow graph models, we choose to compute the commute times between all patches, not just between patches that are connected with an edge.

For each signal model, we compute the square root of the average approximate commute time

$$\sqrt{\kappa'} = \sqrt{\frac{2}{N(N-1)} \sum_{n<m} \|\Phi(\mathbf{x}_n) - \Phi(\mathbf{x}_m)\|^2} \tag{4.22}$$

for a pair of patches, $\mathbf{x}_n$ and $\mathbf{x}_m$, that are either both fast, or both slow patches. We study how κ' varies as a function of the autocorrelation parameter that controls how irregular the fast patches

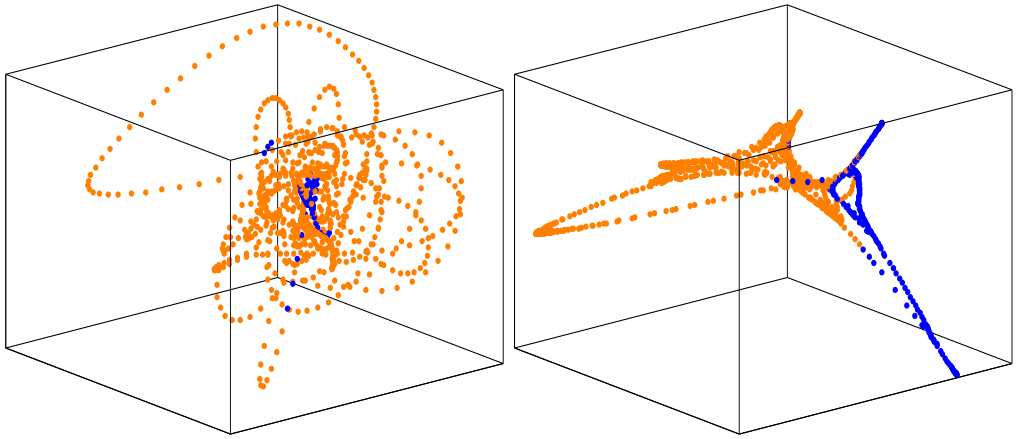

Figure 4.11: Patch-set of the time-frequency signal (see Figure 4.9) before (left) and after (right) embedding. The color-code matches the color used in the plot of the signal: blue = low frequency, orange = high frequency.

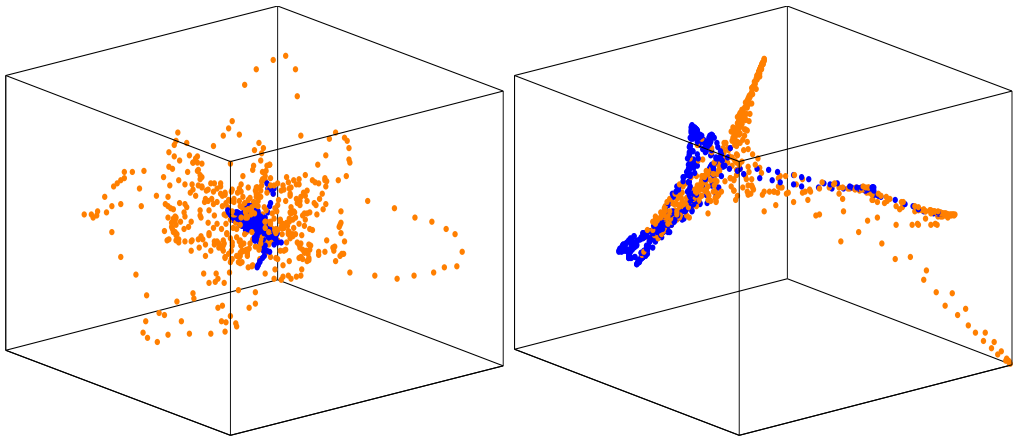

Figure 4.12: Patch-set of the local regularity signal (see Figure 4.10) before (left) and after (right) embedding. The color-code matches the color used in the plot of the signal: blue = smooth, orange = irregular.

are. κ' was computed using ten realizations of each signal model. The slow patches were generated using $\beta_{\mathcal{S}} = 8$ and $H_{\mathcal{S}} = 0.9$. As before, we used $N = 1024$ and $d = 32$ for the time-frequency model and $d = 16$ for the local regularity model. We observed that the overall shapes of the curves in (4.13) is invariant under variation of the parameters (as along as the ratio of the patch length to the average subinterval length remains less than 10%). The dimension d' of the embedding used to

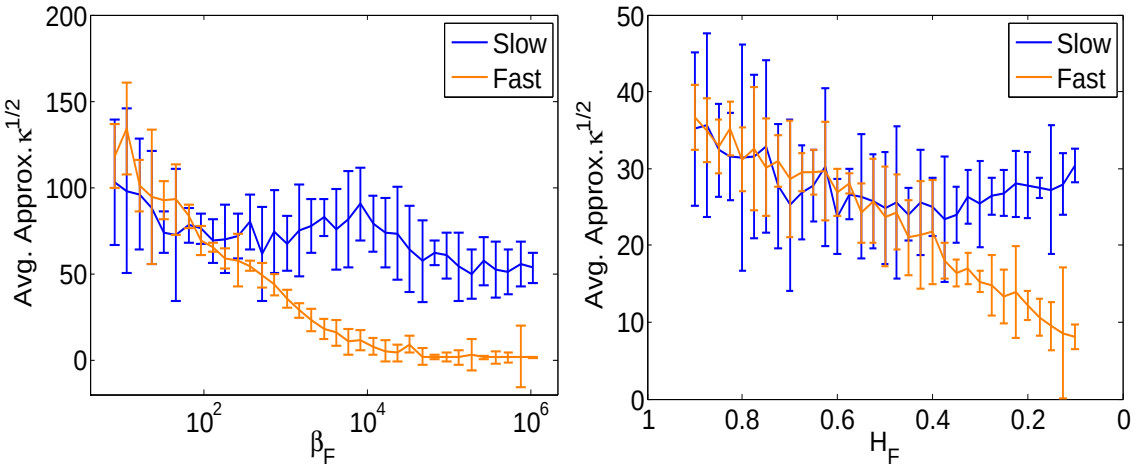

Figure 4.13: $\sqrt{\kappa'}$ for slow (blue) and fast patches (orange) for the time-frequency model (left) and the local regularity model (right) as a function of the "roughness" of the fast patches. The slow patches were generated using $\beta_{\mathcal{S}} = 8$ (left) and $H_{\mathcal{S}} = 0.9$ (right).

compute κ' was chosen so that $(1-\lambda_k)^{-1} < 0.1(1-\lambda_2)^{-1}$, for all $k > d'+1$. Figure 4.13 shows $\sqrt{\kappa'}$ as a function of the frequency parameter (left), and smoothness parameter (right). We note that as the signal exhibits more rapid, local changes (increasing $\beta_{\mathcal{F}}$, or decreasing $H_{\mathcal{F}}$), the associated fast patches are increasingly concentrated (smaller $\|\Phi(\mathbf{x}_n) - \Phi(\mathbf{x}_m)\|$) through the parametrization. These experiments confirm that the theoretical analysis of section 4.4 can be applied to a general patch-set constructed from realistic signals.

4.6 Conclusion

We have confirmed experimentally that embedding the fused graph using Φ shrinks the mutual distance between vertices of the fast subgraph, effectively concentrating these vertices closer to one another. As a result, the embedding helps divide the fused graph into the slow and the fast subgraphs by concentrating the vertices of the fast subgraph away from the vertices of the slow subgraph. Our analysis of the embedding is based on the fact that Φ approximately preserves the commute time measured on the fused graph. Furthermore, we have demonstrated that a truncated version of the commute time, κ', is even more conducive to identifying vertices of the fast subgraph

of the fused graph.

The implication of these results is that the embedding of the true patch-graph Γ using Φ will concentrate the "anomalous" patches, which contain rapid changes in the signal, away from the baseline patches. This concentration of the fast anomalous patches happens for values of the embedding dimension d' that are of the order of $\ln(N)$: this choice of d' results in a low-dimensional embedding of the patch-graph. Because the fast patches are more clustered after embedding, their detection – for the purpose of detection of anomalies, classification, or segmentation – will become much easier. Finally, we note that our theoretical analysis can be extended to a more general context where patches are replaced by a vector of local features extracted from elements of a large dataset. The only requirement is that the graph of features exhibit a geometry similar to the fused graph Γ^*.

Chapter 5

Estimating seismic arrivals

5.1 Introduction

In this section, we propose a new method to analyze seismic time series and estimate the arrival-times of seismic waves, which illustrates the ideas presented in chapters 3 and 4. We validate our approach using a dataset of seismic events that occurred in Idaho, Montana, Wyoming, and Utah between 2005 and 2006. Our approach outperforms methods based on singular-spectrum analysis, wavelet analysis, and the short-term average to long-term average ratio (STA/LTA).

5.1.1 Estimation of arrival-times

Seismic waves arrive at recording stations as distinct bursts, or arrivals, corresponding to different types of motion (e.g. compressional vs. shear) and different propagation paths through the Earth (refracted, reflected, diffracted). Arrival-times of seismic waves are indispensable to the determination of the location and type of the seismic event; the precise estimation of arrival-times remains therefore a fundamental problem. This chapter addresses the problem of estimating the timing of different seismic waves from a seismogram.

Several methods for estimating arrival-times use some variant of the classic current-value-to-predicted-value ratio method (e.g., [4, 67, 28], and references therein). The current value is a short term average (STA) of the energy of the incoming data, while the predicted value is a long term average (LTA), so the ratio is expressed as STA/LTA. This ratio is constantly updated as new data flows in, and a detection is declared when the ratio exceeds a threshold value. When the signal and

the noise are Gaussian distributed, the STA/LTA method yields an optimal detector that strikes the optimal balance between the false alarm rate, or "mispicks" [65] and the missed detections rate [37, 12]. As demonstrated by [70], this theoretical model is unrealistic since seismic waves are non-Gaussian. As a result the seismic waves should be characterized not just by their mean and variance (as in STA/LTA), but also by higher order statistics (such as skewness and kurtosis). These higher order moments have been used to detect the onset of seismic waves [74, 51, 39].

The performance of a detector can be improved by enhancing the signal transients relative to the background noise. Several time-frequency and time-scale decompositions have been proposed for this purpose (e.g., [97, 6, 95] and references therein). Advanced statistical methods can use training data (in the form of seismograms labeled by an analyst). For instance, the software developed at the Prototype International Data Center (Arlington, VA) is based on a multi-layer neural network that uses labeled waveforms in order to predict the types of waves of unseen seismograms [94].

5.1.2 Problem statement

We are interested in detecting seismic waves and estimating the arrival-time of each wave. We model the seismogram $x(t)$ as a sum of two components

$$x(t) = b(t) + w(t), \tag{5.1}$$

where $w(t)$ represents a seismic wave arriving at time τ, and $b(t)$ represents the baseline (or background) activity.

We assume that $b(t)$ models the background noise. In contrast, we expect that $w(t)$ will be a fast oscillatory transient localized around the arrival-time τ (see Figure 5.1). Our goal is to detect the seismic wave $w(t)$, and estimate its onset τ. The difficulty of the problem stems from the fact that there is a large variability in the shape and frequency content of the seismic waves $w(t)$.

We tackle this question by considering the patch-set associated with the model (5.1). As explained in section 5.2, *baseline patches (or slow patches)* that do not overlap with the seismic wave $w(t)$ and only contain the baseline signal $b(t)$ become tightly clustered along low-dimensional

curves. In contrast, *arrival patches (or fast patches)* that include portions of the seismic wave $w(t)$ remain at a large distance from one another, and are also at a large distance from the baseline patches. The differential organization of the baseline and arrival patches in $\mathbb{R}^d$ is the first ingredient of our approach.

This commute time parametrization (4.8) allows us to represent patches from $\mathbb{R}^d$, where d is of the order of 10^3 using only about $d' = 25$ coordinates. Finally, the last stage of our approach consists in training a classifier to detect patches containing seismic waves. The classifier uses the low-dimensional parametrization of the patch-set (4.8).

In summary, the contribution of this chapter is a novel method to analyze seismograms and estimate arrival-times of seismic waves. Our approach includes the following three steps:

(1) Construct the patch-sets associated with each seismogram;

(2) Compute the commute time parametrization using every patch;

(3) Construct a classifier which uses the commute time parametrization; detect the presence of seismic waves and estimate the arrival-times.

5.2 Mutual distance between two patches

In the following, we assume that the seismogram is described by the model (5.1) and we study the Euclidean distance between any two patches $\mathbf{x}(t_n)$ and $\mathbf{x}(t_m)$ extracted at times t_n and t_m. We first consider the case where both patches come from the baseline part of the signal. In this case, we assume $w(t) = 0$ over the intervals $[t_n, t_n + d\Delta t)$ and $[t_m, t_m + d\Delta t)$, and we have

$$\|\mathbf{x}(t_n) - \mathbf{x}(t_m)\|^2 = \sum_{k=0}^{d-1} \left(b(t_n + k\Delta t) - b(t_m + k\Delta t)\right)^2 . \tag{5.2}$$

If the baseline signal varies slowly, then we have $|b(t_n + k\Delta t) - b(t_m + k\Delta t)| \approx 0,\ (k = 0, \ldots, d-1)$, and therefore

$$\|\mathbf{x}(t_n) - \mathbf{x}(t_m)\|^2 = \sum_{k=0}^{d-1} \left(b(t_n + k\Delta t) - b(t_m + k\Delta t)\right)^2 \approx 0. \tag{5.3}$$

We now consider the case where one patch, $\mathbf{x}(t_n)$ (without loss of generality), is part of a seismic wave $w(t)$, whereas the other patch, $\mathbf{x}(t_m)$, comes from the baseline part. We have

$$\begin{aligned}\|\mathbf{x}(t_n)-\mathbf{x}(t_m)\|^2 &= \sum_{k=0}^{d-1}\left(w(t_n+k\Delta t)+b(t_n+k\Delta t)-b(t_m+k\Delta t)\right)^2 \\ &= \sum_{k=0}^{d-1} w^2(t_n+k\Delta t)+2\sum_{k=0}^{d-1} w(t_n+k\Delta t)\left(b(t_n+k\Delta t)-b(t_m+k\Delta t)\right) \\ &\quad +\sum_{k=0}^{d-1}\left(b(t_n+k\Delta t)-b(t_m+k\Delta t)\right)^2.\end{aligned}$$

Again, we can assume that for each k, $|b(t_n+k\Delta t)-b(t_m+k\Delta t)|\approx 0$, and thus

$$\sum_{k=0}^{d-1} w(t_n+k\Delta t)\left(b(t_n+k\Delta t)-b(t_m+k\Delta t)\right)\approx 0. \tag{5.4}$$

As before, we have $\sum_{k=0}^{d-1}\left(b(t_n+k\Delta t)-b(t_m+k\Delta t)\right)^2\approx 0$. We conclude that

$$\|\mathbf{x}(t_n)-\mathbf{x}(t_m)\|^2\approx\sum_{k=0}^{d-1} w^2(t_n+k\Delta t). \tag{5.5}$$

The sum (5.5) measures the energy of the (sampled) seismic wave over the interval $[t_n, t_n+d\Delta t)$. Because the patch size, $d\Delta t$, is chosen so that $w(t)$ oscillates several times over the patch (see Figure 5.1), the interval $[t_n, t_n+d\Delta t)$ is comprised of several wavelengths of $w(t)$, and the energy (5.5) is usually large. Finally, we consider the case where both patches contain part of the seismic wave $w(t)$ (see Figure 5.1),

$$\begin{aligned}\|\mathbf{x}(t_n)-\mathbf{x}(t_m)\|^2 &= \sum_{k=0}^{d-1}\left(b(t_n+k\Delta t)-b(t_m+k\Delta t)\right)^2+\sum_{k=0}^{d-1}\left(w(t_n+k\Delta t)-w(t_m+k\Delta t)\right)^2 \\ &+ 2\sum_{k=0}^{d-1}\left(w(t_n+k\Delta t)-w(t_m+k\Delta t)\right)\left(b(t_n+k\Delta t)-b(t_m+k\Delta t)\right).\end{aligned}$$

If we assume that the baseline signal varies slowly over time, then we have

$$\|\mathbf{x}(t_n)-\mathbf{x}(t_m)\|^2\approx\sum_{k=0}^{d-1}\left(w(t_n+k\Delta t)-w(t_m+k\Delta t)\right)^2. \tag{5.6}$$

The sum (5.6) measures the energy of the difference between two overlapping sections of the seismic wave $w(t)$, sampled every Δt (see Figure 5.1). In order to estimate the size of this energy, we approximate the seismic wave with a cosine function (see Figure 5.1), $w(t)=\cos(\omega t)$, where the

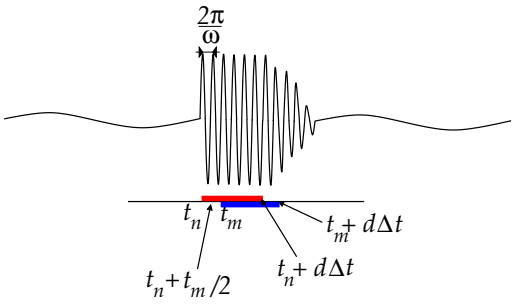

Figure 5.1: The two patches (in red and in blue) contain part of the seismic wave $w(t)$.

frequency ω corresponds to the peak of the short-time Fourier transform of $w(t)$ around τ. In this case, we have

$$\begin{aligned} w(t_n + k\Delta t) - w(t_m + k\Delta t) &= \cos\{\omega(t_n + k\Delta t)\} - \cos\{\omega(t_m + k\Delta t)\} \\ &= 2\sin\{\omega(t_m - t_n)/2\}\sin\{\omega\left[(t_n + t_m)/2 + k\Delta t\right]\}, \end{aligned}$$

and the sum (5.6) becomes

$$\sum_{k=0}^{d-1}\{w(t_n + k\Delta t) - w(t_m + k\Delta t)\}^2 = 4\sin^2\{\omega(t_m - t_n)/2\}\sum_{k=0}^{d-1}\sin^2\{\omega\left[(t_n + t_m)/2 + k\Delta t\right]\}. \tag{5.7}$$

The sum in the right-hand side of (5.7) can be written as

$$\sum_{k=0}^{d-1}\sin^2\{\omega\left[(t_n + t_m)/2 + k\Delta t\right]\} = \sum_{k=0}^{d-1}\cos^2\left\{\omega\left[(t_n + t_m)/2 + \frac{\pi}{2\omega} + k\Delta t\right]\right\}. \tag{5.8}$$

The right-hand side of (5.8) is the energy of the cosine function sampled every Δt on an interval starting at $(t_n + t_m)/2 + \pi/2\omega$ of length $d\Delta t$ (see Figure 5.1). As explained above, $d\Delta t \gg 2\pi/\omega$, and thus the energy (5.8) is measured over several wavelengths and is therefore large. Finally, given a patch starting at time t_n, all patches starting at time t_m, where $t_m = t_n + (q + 1/2)2\pi/\omega$, for some $q = 0, 1, \ldots$ will satisfy $\sin^2(\omega(t_m - t_n)/2) = 1$. We conclude that given a patch starting at time t_n, there are many choices of t_m such that the sum in (5.7), and therefore the norm in (5.6), are large.

In summary, we expect the mutual distance between patches extracted from the baseline signal to be small, while the mutual distance between baseline and arrival patches will be large. Moreover, we also expect that two arrival patches will often be at a large distance of one another.

This organization between the seismic patches is similar to the type of organization between fast and slow patches that is studied in chapter 4. Therefore, parametrizing the seismic patches using (4.8) is expected to concentrate those patches that contain the seismic arrivals.

5.3 Normalization of the patch-set

We now consider the following question: if we want to use seismograms from different stations to learn the general shape of a seismic wave, how should we normalize the seismograms? The magnitude of an earthquake, which characterizes its damaging effect, is defined as a logarithmic function of the radiated energy [72]. The radiated energy can be estimated by integrating the velocity associated with the displacement measured by seismograms [8]. A logarithmic normalization would make it possible to account for the large variability in the energy and would allow us to compare seismograms from different stations or from different events. We favor an equivalent normalization that consists in rescaling each patch by it energy. More precisely, for each patch $\mathbf{x}(t)$ at a fixed time t, we center the patch using (3.12), and then fix the Euclidean norm with (3.13).

5.4 Estimation of Arrival-Times of Seismic Waves

5.4.1 Learning the presence of seismic waves in the patch-set

Our goal is to learn the association between the presence of a seismic wave within a patch, and the values of the patch coordinates. As explained before, we advocate a geometric approach: we expect that patches will organize themselves on the unit sphere in $\mathbb{R}^d$ in a manner that will reveal the presence of seismic waves. We represent all the patches with the coordinates defined by Φ in (4.8). We then use training data (labeled by experts) to partially populate the patch-set with information about the presence or absence of seismic waves. We combine the information provided

by the labels with the knowledge about the geometry of the patch-set to train a classifier; this approach is known as semi-supervised learning [18]. We then use the classifier to classify unlabeled patches into baseline, or arrivals patches. The classification problem is formulated as a kernel ridge regression problem [47]: for any given patch, the classifier returns a number between 0 and 1 that quantifies the probability that a seismic wave be present within the patch.

We assume that N_l of the N patches have been labeled by an expert (analyst): for each of these patches we know if a seismic wave was observed in the patch, and at what time. We construct a response function f defined on the new coordinates, $\Phi(\mathbf{x}_n) \in \mathbb{R}^{d'}$, and taking values in $[0,1]$,

$$f : \mathbb{R}^{d'} \longrightarrow [0,1] \tag{5.9}$$

$$\Phi(\mathbf{x}_n) \longrightarrow f(\Phi(\mathbf{x}_n)). \tag{5.10}$$

The range $[0,1]$ is arbitrary: 0 is the absence of a response, while 1 is the maximum response. The classifier decides that the patch $\mathbf{x}_n$ contains an arrival if the response $f(\Phi(\mathbf{x}_n))$ is greater than some threshold $\varepsilon > 0$. The threshold ε controls the rates of false alarms and missed detections: a small ε results in many false alarms but will rarely miss arrivals, and vice versa. We expand the response function as a linear combination of Gaussian kernels in $\mathbb{R}^{d'}$,

$$f(\Phi(\mathbf{x})) = \sum_{j=1}^{N_l} \beta_j \, \exp\left\{-\|\Phi(\mathbf{x}) - \Phi(\mathbf{x}_j)\|^2/\alpha^2\right\}. \tag{5.11}$$

The vector of unknown coefficients $\boldsymbol{\beta} = \left(\beta_1, \ldots, \beta_{N_l}\right)$ is computed using the training data. The kernel ridge regression [47] combines two ideas: distances between patches are stored in the Gaussian kernel matrix $\boldsymbol{K}$, with entries $\mathbf{K}_{n,m} = \exp\left\{-\|\Phi(\mathbf{x}_m) - \Phi(\mathbf{x}_n)\|^2/\alpha^2\right\}$, $n, m = 1, \ldots, N_l$; and the classifier is designed to provide the simplest model of the response in terms of the N_l training data. Rather than trying to find the optimal fit of the function f to the N_l labeled patches, we penalize the regression (5.11) by imposing a penalty on the norm of $\boldsymbol{\beta}$ [47]. This prevents the model (5.11) from overfitting the training samples. The optimal regression is defined as the solution to the quadratic minimization problem

$$\|\boldsymbol{r} - \boldsymbol{K}\boldsymbol{\beta}\|^2 + \mu \boldsymbol{\beta}^T \boldsymbol{K} \boldsymbol{\beta}, \tag{5.12}$$

where $\|\cdot\|$ is the Euclidean norm on $\mathbb{R}^{N_l}$, and $\boldsymbol{r} = \left(r_1, \ldots, r_{N_l}\right)^T$ is the known response for the N_l labeled patches. The parameter μ controls the amount of penalization: $\mu = 0$ yields a least squares fit, while $\mu = \infty$ ignores the data. For a given choice of μ, the optimal vector of coefficients [47] is given by

$$\boldsymbol{\beta} = (\boldsymbol{K} + \mu \boldsymbol{I})^{-1} \boldsymbol{r}, \tag{5.13}$$

where $\boldsymbol{I}$ is the $N_l \times N_l$ identity matrix. In our experiments, the ridge parameter μ was determined by cross-validation, and the same value, $\mu = 0.8$, is used throughout. The Gaussian width α is chosen to be a multiple of the average kernel distance,

$$\alpha^2 = C \ \left(\text{average}_{\mathbf{x}_n, \mathbf{x}_m} \|\Phi(\mathbf{x}_n) - \Phi(\mathbf{x}_m)\|^2\right); \tag{5.14}$$

for all experiments we choose $C = 0.51$.

5.4.2 Defining ground truth

5.4.2.1 Uncertainty in arrival-time

In order to validate our approach we need to compare the output of the response function f, defined by (5.11), to the actual decision provided by an expert (analyst). The comparison is performed for every patch being analyzed. Before we present the result of the comparison (see section 5.5.2) we need to properly define the ground truth. The decision of the analyst is usually formulated as a binary response: an arrival is present at time τ_i or not. We claim that this apparent perfect determination of the arrival-time is misleading. Indeed, [36] argues that the origin time and the arrival-time at a given station are, "for all practical purposes, random variables whose distributions" depend on the quality of the seismic record and the training and experience of the analyst detecting the arrivals. We formalize this intuition and model the arrival-time estimated by the analyst as a Gaussian distribution with mean τ_i and variance h_i. The parameter h_i controls the width of the Gaussian and quantifies the confidence with which the analyst estimated τ_i. Ideally, h_i should be a function of the inter-observer variability for the estimation of τ_i. In this work, we propose to estimate the uncertainty h_i directly from the seismogram. For each arrival-time τ_i, we

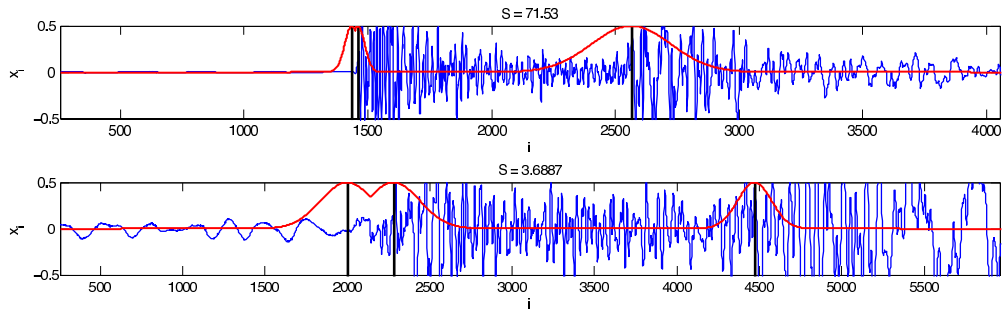

Figure 5.2: Seismic traces x_n (blue); estimated responses r_n (red); arrival-times τ_i (black vertical bars).

compute the dominant frequency ω of $x(t)$ using a short-time Fourier transform. Let $T = 1/\omega$ be the period associated with ω, we define the uncertainty h_i as follows

$$h_i = \begin{cases} 2T/\Delta t & \text{if } 2T/\Delta t < h_{\max}, \\ h_{\max} & \text{otherwise.} \end{cases} \tag{5.15}$$

This choice of h_i corresponds to the following idea: if the seismogram were to be a pure sinusoidal function oscillating at the frequency ω (see Fig 5.1), then this choice of h_i would guarantee that we observe two periods (cycles) of $x(t)$ over a time interval of length h_i.

Finally, we define the true response r_n at time t_n to be the maximum of the Gaussian bumps associated with the arrival-time times τ_i nearest to time t_n,

$$r_n = \max_i \left\{ \exp\left(-(t_n - \tau_i)^2 / h_i \right) \right\}. \tag{5.16}$$

Figure 5.2 displays two seismograms with different values for h_i. In the top seismogram the first two arrivals are very localized (small h_1 and h_2), whereas the third arrival corresponds to a lower instantaneous frequency, and is therefore less localized (large h_3). In the second seismogram (bottom of Figure 5.2) the first two arrivals are very close to one another resulting in an overlap of the Gaussians defining the response r_n.

5.4.2.2 Energy localization of seismic traces

Because we analyze seismic traces of very different quality, we need to find a way to quantify these differences so we can assess the corresponding variability in the response of our proposed methodology. It is well-known that variability in the estimation of arrival-times by an analyst is less pronounced when a seismic trace contains very localized arrivals [90], hence we propose using energy localization as our metric to categorize waveforms. We define the energy localization of a given trace to be the average ratio of the energy of the seismic waves present in the trace over the energy of the baseline activity. This is related to the STA/LTA ratio, but differs in that we form a single statistic for an entire waveform rather than a transformed time series. For a given trace let $\mathcal{A}$ be the subset of patches that contain arrivals, and $\mathcal{B}$ be the complement of $\mathcal{A}$, i.e. the subset of patches that contain only baseline activity. We define the energy localization by the ratio

$$S = \frac{\sum_{\mathcal{A}} \|\mathbf{x}_n\|^2/|\mathcal{A}|}{\sum_{\mathcal{B}} \|\mathbf{x}_n\|^2/|\mathcal{B}|}, \tag{5.17}$$

where $|\mathcal{A}|$ and $|\mathcal{B}|$ are the numbers of patches in $\mathcal{A}$ and $\mathcal{B}$, respectively. Figure 5.3 shows two seismic traces with very different energy localizations ($S = 26.0$ vs. $S = 1.3$). Arrival-times assigned by an analyst are represented by vertical bars, and STA/LTA processed time series are shown for comparison.

5.4.3 Optimization of the STA/LTA parameters

For STA/LTA processing, we first apply a 0.8-3.5 Hz bandpass Butterworth filter to enhance the SNR. Instead of using fixed window sizes, we adaptively optimize the window sizes. The optimal sizes for the short and long windows were selected using a procedure similar to the procdeure used in [65]. As in [65], the goal of the optimization is to minimize the number of false alarms ("mispicks") while reducing the number of missed detections ("highest accuracy"). We quantify this optimality criterion with the Receiver Operating Characteristic (ROC) curve [47], a standard procedure in statistics. The ROC curve is a plot of the true detection rate – which quantifies the accuracy of the detector, as a function of the false alarm rate – which quantifies the rate of mispicks. In order

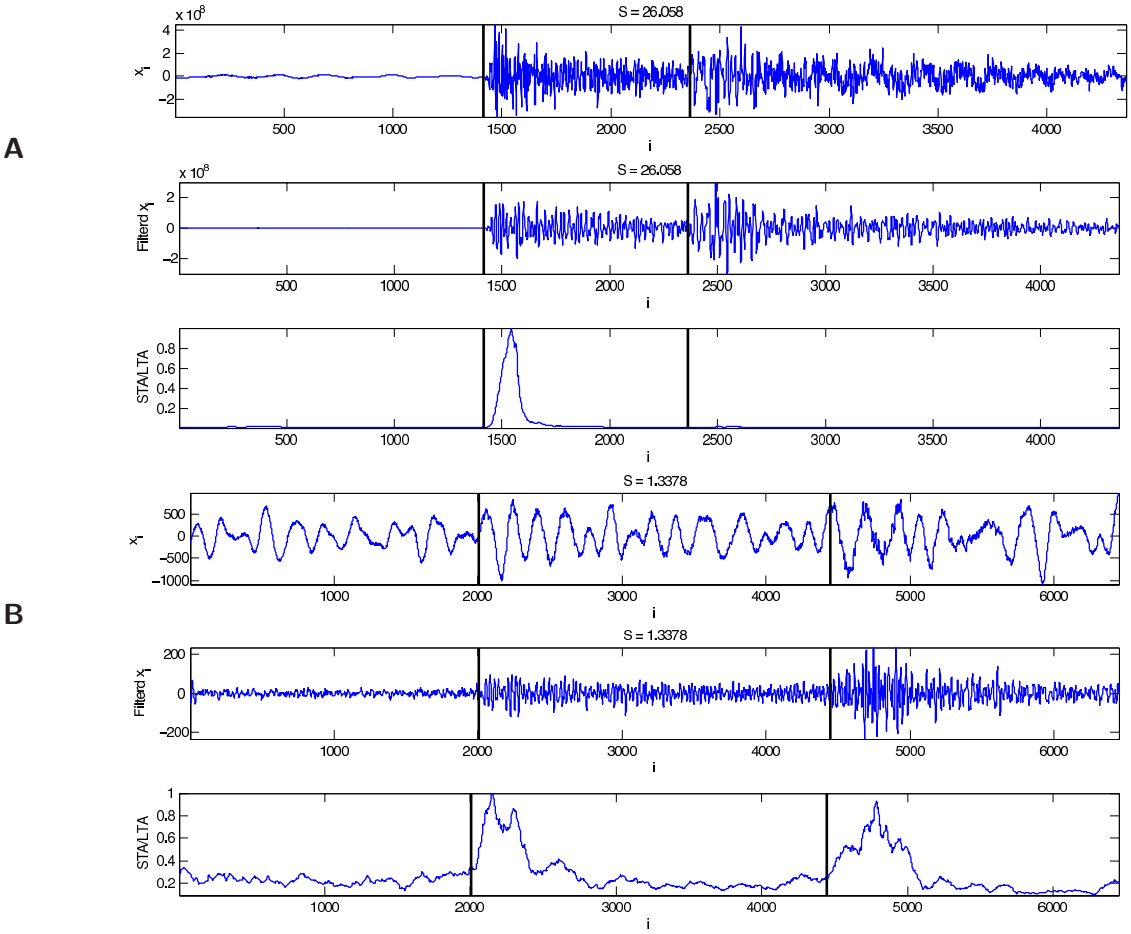

Figure 5.3: Raw and filtered seismic traces with associated STA/LTA outputs. **A**: high energy localization ($S = 26.0$) and **B**: very diffuse energy localization ($S = 1.3$). Analyst picks are represented by bars.

to provide a summary of the entire curve, we compute the area under the curve: the closer the area is to one, the better the accuracy of the dectector.

Different sizes were selected for the three levels of energy localization ratio S in the Rocky Mountain data set. The optimal short window was selected in the range $[1.6\text{ s}, 25.6\text{ s}]$ and the optimal long window was selected in the range $[3.2\text{ s}, 51.2\text{ s}]$. There was no delay between the windows. As in [65] we use part of the data to compute the optimal STA/LTA window sizes, and we then use the remaining traces to evaluate the performance of STA/LTA. All the results that are reported in the STA/LTA experiments were obtained after optimizing the parameters. We found

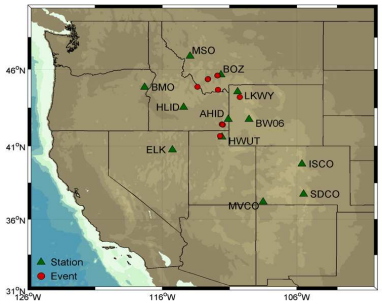

Figure 5.4: Locations of the stations and events from the Rocky Mountain region.

that the optimal short/long window sizes were: 1.6/3.2 s, 1.6/51.2 s, and 12.8/25.6 s for the low, medium, and high localization ratio S, respectively.

For the purpose of comparison, we normalized the STA/LTA output so that its maximum value is one. The trace (**A**) in Figure 5.3 has a large energy localization, while the trace (**B**) has a very low energy localization. The Butterworth filter is able to remove some of the irrelevant low-frequency oscillations in (**B**) and yields a signal that can be processed by STA/LTA. We note that the second arrival (Lg) in the first trace (**A**) is missed by the STA/LTA algorithm. We believe that STA/LTA missed the Lg arrival because the interval between the two arrivals is shorter than the LTA window length so the LTA cannot be reset properly. In other words, the primary arrival is still in the LTA window when the secondary arrival moves into the STA window.

5.5 Results

5.5.1 Rocky mountain dataset

We validate our approach with a dataset composed of broadband seismic traces from seismic events that occurred in Idaho, Montana, Wyoming, and Utah between 2005 and 2006 (see Figure 5.4). While the data set is small, it provides a set of wave propagation paths and recording station environments that is broad enough to validate our new algorithm. Arrival-times have been determined by an analyst. The ten events with the largest number of arrivals were selected for analysis. In total, we used 84 different station records from ten different events containing 226

labeled arrivals. Of the 226 labeled arrivals, there are 72 Pn arrivals, 70 Pg arrivals, 6 Sn arrivals, and 78 Lg arrivals. The sampling rate was $1/\Delta t = 40$ Hz. We consider only the vertical channel in our analysis. To minimize the computational cost, patches are spaced apart by $40\Delta t$ (1 s).

5.5.2 Validation of the classifier

The performance of the algorithm varies as a function of the energy localization S, and therefore we perform three independent validations by dividing the seismic traces into three homogeneous subsets: $n_1 = 27$ traces with low energy localization ($S < 3$), $n_2 = 29$ traces with medium energy localization ($3 \leq S \leq 18$), and the remaining $n_3 = 28$ traces with high energy localization ($S > 18$).

For comparison purposes, we also processed the data set with the optimized STA/LTA, as described in section 5.4.3.

Figures 5.5, 5.6, and 5.7 show twenty seven seismic traces that are representative of the three energy localization subsets: high, medium and low, respectively. STA/LTA (magenta) always misses the secondary wave for medium and high energy localizations, and often misses the secondary wave at low energy localization level. Our approach (green) can detect all primary and secondary waves at high and medium energy localization levels. At low localization levels, our approach sometimes yields too early a detection.

For each subset s, $(s = 1, 2, 3)$, we perform a standard leave-one-out cross-validation [47] using n_s folds as follows. We choose a test seismogram $x_{test}(t)$ among the n_s traces and compute the optimal set of weights (5.13) for the kernel ridge classifier (5.11) using the remaining $n_s - 1$ traces. Patches $\mathbf{x}_n$ are then randomly selected from the test seismogram $x_{test}(t)$ and the classifier computes the response function $f(\Phi(\mathbf{x}_n))$. The response of the classifier is compared to the true response r_n for various false alarms and missed detections levels. We repeat this procedure for each possible test seismogram $x_{test}(t)$ among the n_s seismograms. Figure 5.8 details the cross-validation procedure. We quantify the performance of the classifier using a Receiver Operating Characteristic (ROC) curve [47]. The true detection rate is plotted against false alarm rate. We characterize each

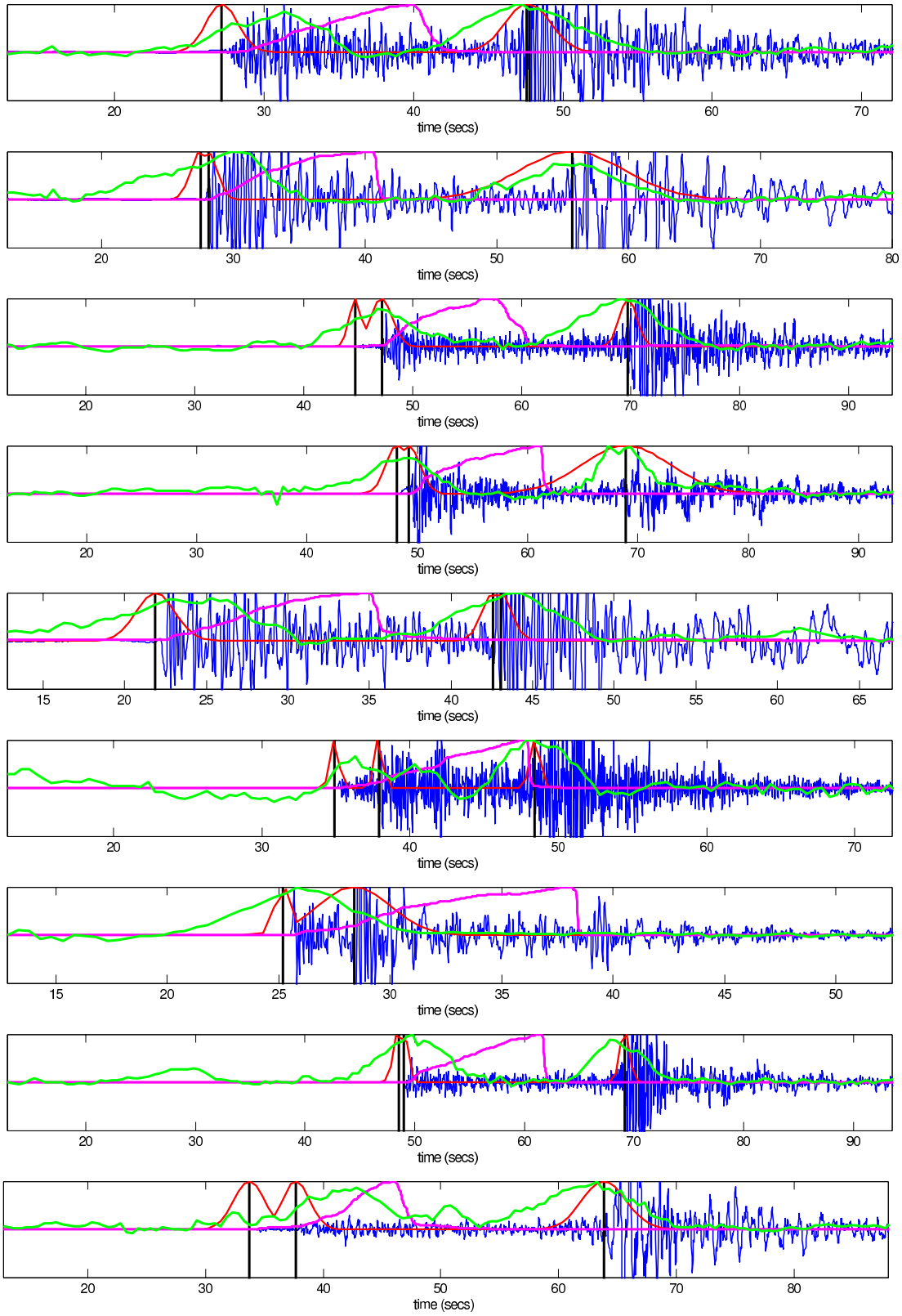

Figure 5.5: Example output from high-S partition. Seismic trace x_n (blue); true response r_n (red); STA/LTA (magenta); classifier $f(\Phi(\mathbf{x}_n))$ (green).

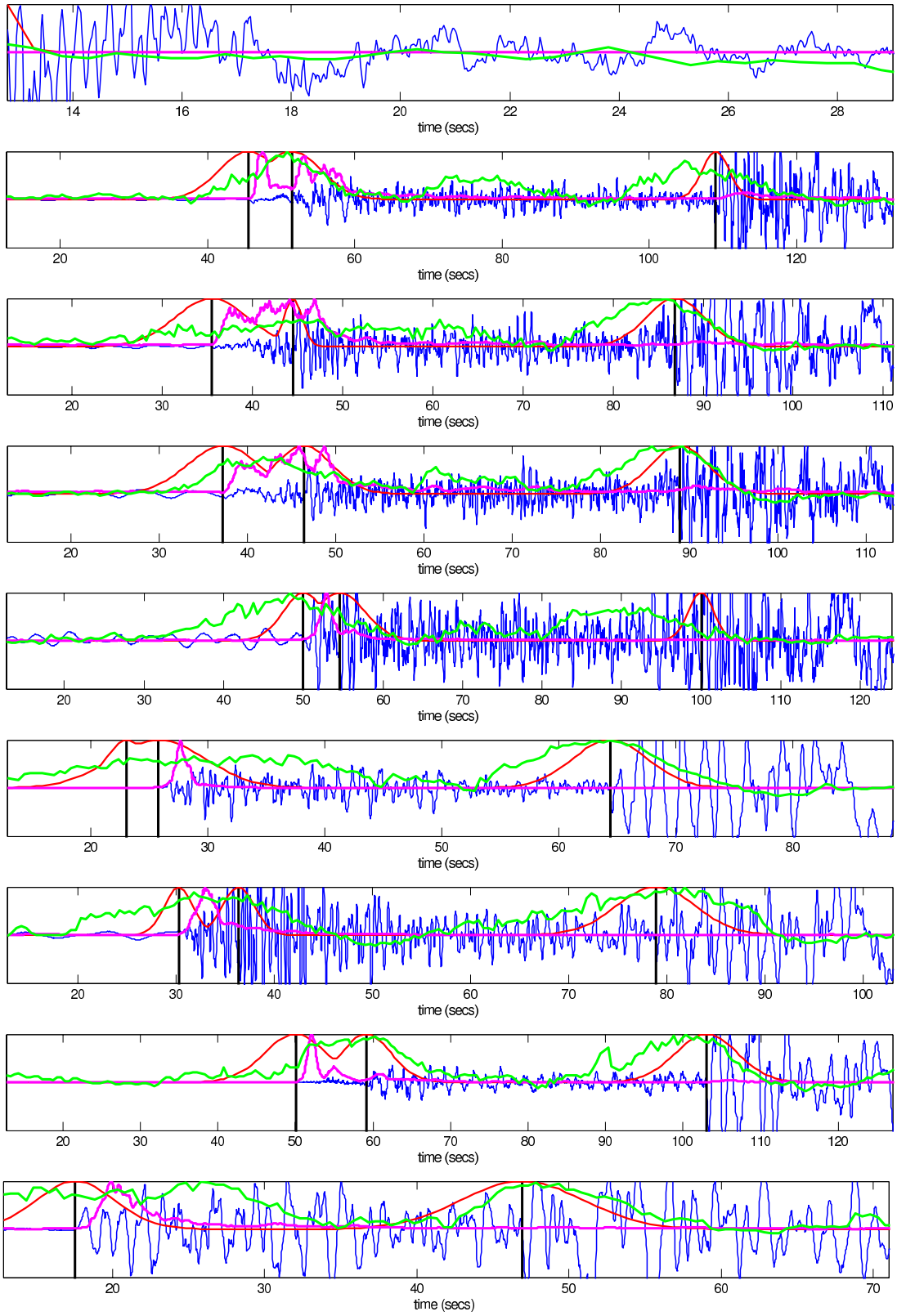

Figure 5.6: Example output from medium-S partition. Seismic trace x_n (blue); true response r_n (red); STA/LTA (magenta); classifier $f(\Phi(\mathbf{x}_n))$ (green).

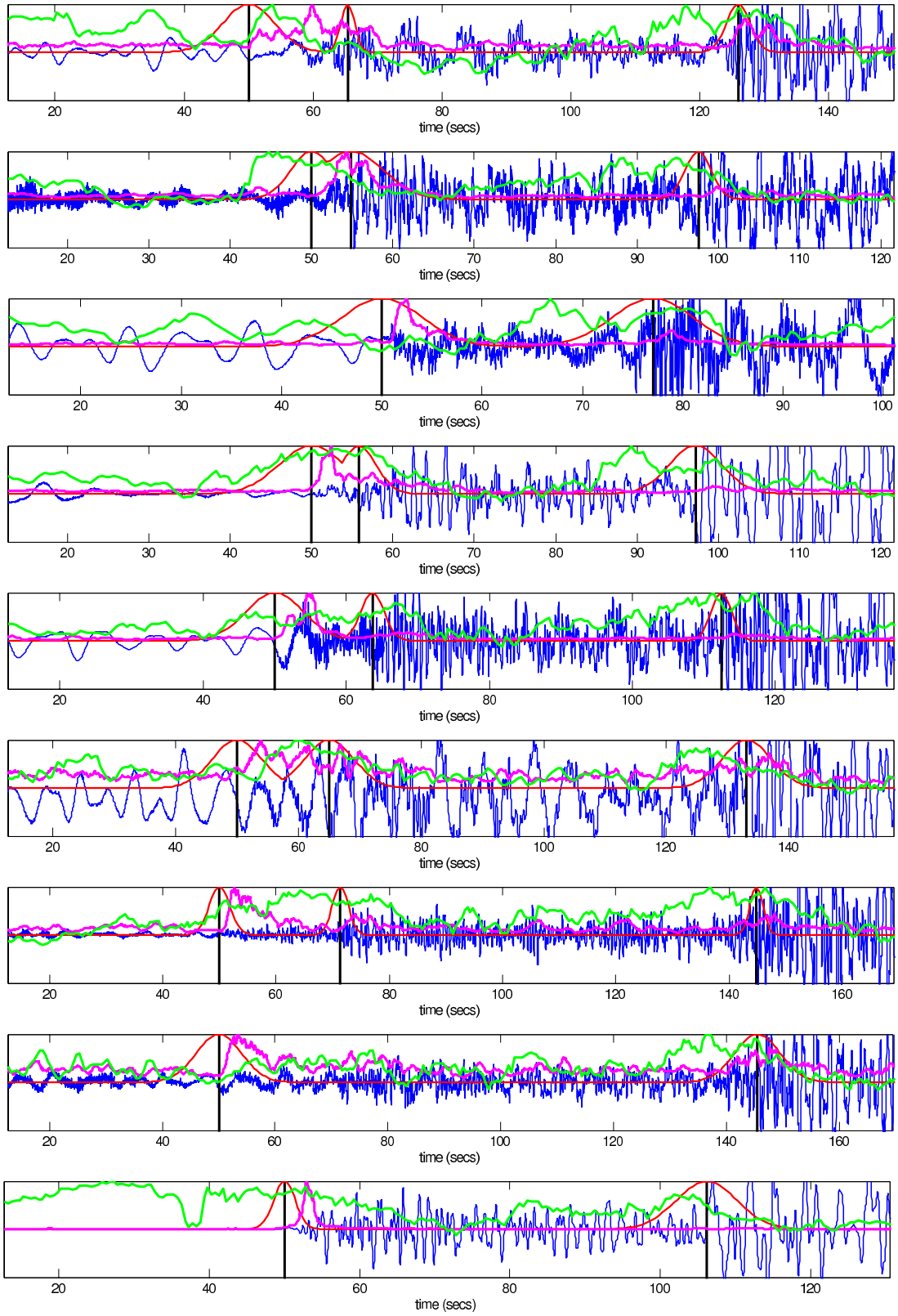

Figure 5.7: Example output from low-S partition. Seismic trace x_n (blue); true response r_n (red); STA/LTA (magenta); classifier $f(\Phi(\mathbf{x}_n))$ (green).

ROC curve by the area under the curve (the closer to one, the better).

5.5.3 Optimization of the parameters of the algorithm

The optimal values of the parameters were computed using cross-validation. This procedure turned out to be very robust, since we used the same parameters for all experiments. The optimal classification performance was achieved by choosing $\sigma = \infty$ and $\nu_{NN} = 32$ in the construction of the graph Laplacian. This is equivalent to setting the weights $\mathbf{W}_{n,m}$ on the edges to be 1, and yields a graph that is extremely robust to noise. The influence of the patch size on the classification performance can be found in a series of ROC curves in Figure 5.9. We observe in Figure 5.9 that the STA/LTA algorithm performs best for seismograms with low energy localization.

We know that STA/LTA cannot trigger unless the energy level in the LTA window has established a stable level before the signal comes in the STA window at a higher level. This is always more problematic for secondary arrivals, a well-known problem for STA/LTA. For the mid and high energy localization partitions, we observe that often the secondary arrivals either arrive before the primary arrival has left the LTA window or before the energy level after the first arrival has settled down (i.e. the secondary arrival is within the coda of the first arrival). Either way, the LTA is too high for a trigger. This problem can be addressed by making LTA shorter, but STA gets shorter too and the STA/LTA output is less stable. For seismic traces with low energy localization our approach cannot compete with STA/LTA when the patch size drops below 6.4 s, ($d < 256$). Of course, it is unfair to compare our approach using patches of only 3.2 s ($d = 128$) when the optimized STA/LTA, typically uses a combination of 25.6 s for the long window and 1.6 s for the short window. Indeed, as soon as the window size is larger or equal to 25.6 s ($d \geq 1024$), our approach outperforms STA/LTA. Interestingly, our approach does not benefit from using a much larger patch size; when the patch size becomes 51 s ($d = 2048$) the scale of the local analysis is no longer adapted to the physical process that we study.

Algorithm 2: Cross validation of the classification

Input: Seismic traces, and the associated responses (r_n).

Algorithm:

for $s = 1$ **to** 3 *// for each subset of seismic traces*

extract a total of N patches from n_s distinct seismic traces.

compute new coordinates $\Phi(\mathbf{x}_n)$ of each patch $\mathbf{x}_n$ $i = 1, \ldots, N$

for $j = 1$ **to** n_s *// evaluate the classifier for each seismic trace j*

build classifier using all patches except those from trace j

for all patches $\mathbf{x}_n^j$ in trace j

compute classifier response $f(\Phi(\mathbf{x}_n^j))$

for $\varepsilon = \varepsilon_l$ **to** ε_u *// populate the ROC curve using different thresholds to detect an arrival*

if $f(\Phi(\mathbf{x}_n^j)) > \varepsilon$ **and** $r_n < \varepsilon_0$ **then**

declare false positive

else if $f(\Phi(\mathbf{x}_n^j)) < \varepsilon$ **and** $r_n > \varepsilon_0$ **then**

declare false negative

end if

end for

end for

record false positive and false negative rates for patches in fold j

end for

compute average false positive and false negative rate

end for

Output: area under the ROC curve.

Figure 5.8: Cross validation procedure.

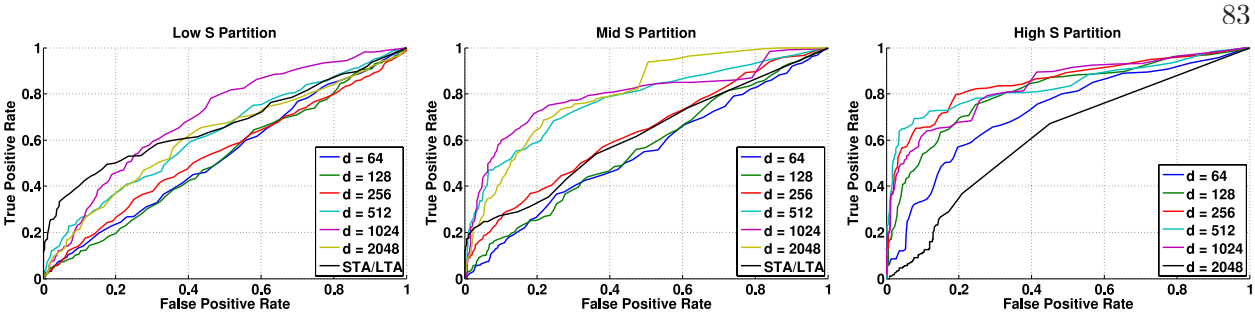

Figure 5.9: ROC curves for various values of the embedding dimension d at three levels of energy localization.

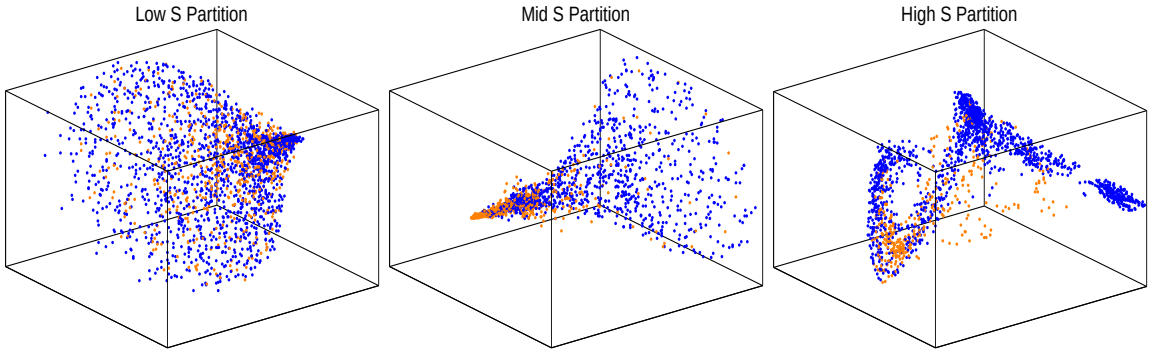

Figure 5.10: Scatter plot of patch-set through the map Φ (4.8), where $d' = 3$. The color encodes the presence (orange) or absence (blue) of an arrival within $\mathbf{x}_n$. The energy localization levels increases from left to right.

5.5.4 So what does the set of patches look like?

To help us gain some insight into the geometric organization of patch space we display the patches using some of the new coordinates $\Phi(\mathbf{x}_n)$. For the three subsets of patches (classified according to the energy localization), we display in Figure 5.10 each patch through the map Φ (4.8), where $d' = 3$. The color of the dot encodes the presence (orange) or absence (blue) of an arrival within $\mathbf{x}_n$. As the energy localization increases the separation between baseline patches and arrival patches increases. This visual impression is confirmed using the quantitative evaluation performed with the ROC curves (see Figure 5.9). Clearly the shape of the set of patches is not linear, and would not be well-approximated with a linear subspace.

5.5.5 Classification performance

We first compare our approach to the gold-standard provided by STA/LTA. The second stage of the evaluation consists in quantifying the importance of the nonlinear dimension reduction Φ defined by (4.8). To gauge the effect of Φ we replace it by two transforms: a wavelet transform and a PCA transform. In both cases, we reduce the dimensionality of each patch from d to d'. Wavelets have been used for a long time in seismology because seismograms can be approximated with very high precision using a small number of wavelet coefficients (e.g., [5, 97, 42] and references therein). On the other hand, we can also try to find the best linear approximation to a set of N patches. This linear approximation is obtained using PCA (also known as the singular-spectrum analysis [89]). The first d' vectors of a PCA analysis yields the subspace that provides the optimal d'-dimensional approximation to the set of patches.

5.5.5.1 STA/LTA ratio

As discussed previously in section 5.4.3, we implement STA/LTA processing with optimized short and long window sizes and a Butterworth 0.8-3.5 Hz band pass preprocessing filter. An arrival is declared when the output exceeds a threshold, which we vary to create the ROC curve.

5.5.5.2 PCA and wavelet representations of the patch-set

An orthonormal wavelet transform (symmlet 8) provides a multiscale decomposition of each patch $\mathbf{x}_n$ in terms of d coefficients. Many of the coefficients are small and can be ignored. In order to decide which wavelet coefficients to retain, we select a fixed set of $d'/2$ indices corresponding to the largest coefficients of the baseline patches. Similarly, we select the $d'/2$ indices associated to the largest coefficients among the patches that contain arrivals. This procedure allows us to define a fixed set of d' wavelet coefficients that are used for all patches as in input to the ridge regression algorithm. Similarly, we keep the first d' coordinates returned by PCA.

Table 5.1: Area under the ROC curve as a function of the patch dimension d and the reduced dimension d', at three different energy localization levels S. The red values correspond to the largest area under the ROC curve for a given feature and dimension d'.

	d	STA/LTA	Wavelet		PCA		Laplacian	
d'			25	50	25	50	25	50
Low S	64	–	0.53	0.53	0.51	0.55	0.53	0.53
	128	–	0.55	0.49	0.52	0.51	0.52	0.52
	256	–	0.52	0.47	0.51	0.54	0.54	0.57
	512	–	0.53	0.50	0.61	0.61	0.62	0.64
	1024	0.68	0.43	0.38	0.54	0.64	0.70	0.71
	2048	–	0.45	0.39	0.55	0.48	0.61	0.62
Mid S	64	–	0.54	0.52	0.52	0.54	0.57	0.57
	128	–	0.57	0.55	0.53	0.56	0.66	0.66
	256	–	0.61	0.62	0.70	0.71	0.71	0.71
	512	–	0.68	0.67	0.76	0.79	0.79	0.81
	1024	0.68	0.77	0.76	0.81	0.84	0.86	0.86
	2048	–	0.64	0.66	0.69	0.75	0.80	0.80
High S	64	–	0.56	0.62	0.65	0.65	0.72	0.67
	128	–	0.68	0.69	0.78	0.73	0.80	0.79
	256	–	0.72	0.70	0.77	0.84	0.88	0.87
	512	–	0.74	0.79	0.73	0.85	0.90	0.90
	1024	0.65	0.72	0.76	0.67	0.75	0.88	0.89
	2048	–	0.51	0.49	0.57	0.67	0.76	0.74

5.5.5.3 Parameters of the classifier based on the PCA and wavelet representations

After applying a wavelet transform, or PCA, we use a ridge classifier (see section 5.4.1) to detect arrivals. The parameters of the classifier are optimized for the wavelet and PCA transforms, respectively. The Gaussian width α was again chosen to be a multiple of the average kernel distance between any two patches (see (5.14)). The parameter C in equation (5.14) was set to $C = 6.9$ for the wavelet parametrization, and $C = 4.6$ for the PCA parametrization. The ridge regression parameter was the same for both wavelets and PCA and was equal to $\mu = 10^{-3}$.

5.6 Conclusion

Table 5.1 provides a detailed summary of the performance of our approach. For each energy localization level (see section 5.5.2 for the definition of the subsets), we report the performance of the different detection methods as a function of d (patch dimension) and d' (reduced dimension).

The performance is quantified using the area under the ROC curve (the ROC curves are shown in Fig. 5.9); a perfect detector should have an area equal to one. The red values in the table correspond to the most accurate detection rate for a given feature and dimension d'.

5.6.1 Effect of the patch size

As expected, the patch-based methods perform poorly if the patch is too small (there is not enough information to detect the seismic wave) or too large (the information is smeared over too large a window). The choice of the optimal patch size is dictated by the physical processes at stake here, since the optimal size is the same for all methods, irrespective of the transform used to reduce dimensionality. For high energy localization seismograms, the seismic waves are very localized and therefore all algorithms perform better with smaller patches: 6.4 s ($d = 256$) or 12.8 s ($d = 512$) instead of 25.6 s ($d = 1024$).

5.6.2 Effect of the transform used to reduce dimensionality

The experiments indicate that PCA outperforms a wavelet decomposition at every energy localization level. Both PCA and the wavelet transform are orthonormal transforms that can be understood in terms of a rotation of $\mathbb{R}^d$. PCA provides the optimal rotation to align the patch-set along the d'-dimensional subspace of best-fit. Finally, the nonlinear transformation Φ based on the eigenfunctions of the Laplacian outperforms both PCA and wavelets. This clearly indicates that the set of patches contains nonlinear structures that cannot be well approximated by the optimal linear subspace computed by PCA. Interestingly, the results (not shown) are not improved by applying a wavelet transform before applying the nonlinear map Φ (4.8) (see [76] for an example of a combination of wavelet transform with a nonlinear map similar to Φ).

5.6.3 Dimension of the patch-set

Notice that the performance is not significantly improved when 50 coordinates are used instead of 25. This is a result that is independent of the method used to reduce the dimensionality,

and is therefore either *(i)* a statement about the complexity of the patch-set and about the physical nature of the seismic traces, or *(ii)* a consequence of some mechanism similar to that described in section 4.4.

As mentioned before, several studies have estimated the dimensionality of the low-dimensional inertial manifold reconstructed from the phase space of the tremors of a single volcano. This dimensionality was found in most studies [26, 27, 49] to be less than five: a number much smaller than our rough estimate of the dimensionality of the patch-set. Because the patch-set includes several seismograms from different events measured at different stations, we expect the dimensionality of this set to be greater than the dimensionality of the phase space reconstructed from the tremors of a single volcano measured at a single station. Yet, our study confirms that the combined phase spaces associated with regional seismic waves remains remarkably low-dimensional. So, it is possible our theoretical conclusions of section 4.4 may account for this behavior.

Chapter 6

Evaluation of fast methods for computation

6.1 Introduction

The commute time parametrization of the patch-graph, given in (4.8), entails two computational bottlenecks. The first computationally expensive procedure is computing nearest neighbors. The second intensive computation is computing the largest eigenvalues λ_k and corresponding eigenvectors ϕ_k, for $k = 1, 2, \ldots, d'$ of the N-by-N matrix $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$. This leads to the following questions. First, how do we quickly compute nearest neighbors? Second, how can we efficiently compute the eigenpairs when N is large? Third, if the size of the patch-set changes, do we need to recompute the eigenpairs? In the following, we discuss some answers to these questions.

6.2 Computing ν nearest neighbors

As the number of points in a dataset gets large, the nearest-neighbor calculation used to build the graph and the calculation of the eigenfunctions can dominate the computational cost. A naive approach to the nearest-neighbor calculation requires $\mathcal{O}(N^2)$ computations. Fortunately, because we choose the number of nearest-neighbors $\nu = \mathcal{O}(\log(N))$ (see chapter 4), the number of nonzero entries in the N-by-N matrix $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ is $\mathcal{O}(N \log(N))$. Consequently, computing the nearest neighbors is of the same order. The cost of the ν nearest neighbor calculation can also be decreased to subquadratic using spatial data structures as described in [48]. In practice, we compute the nearest-neighbor information using the Approximate Nearest Neighbors Toolbox [64].

6.3 Out-of-sample extension

Denote the set of N patches as $\overline{X}$. When N is large, we compute the eigenpairs (λ_k, ϕ_k) using a small subset of patches, which we denote as $X \subset \overline{X}$. Then, we extend the eigenfunctions to the remaining patches. When dealing with the eigenfunctions ϕ_k, this approach is also known as the Nyström extension [33]. In this section, we introduce the Nyström extension in the context of interpolation using radial basis functions (RBFs).

We assume that the patches in X are denoted by $\mathbf{x}_i$ for $i = 1, 2, \ldots, M$ and the remaining $N - M$ patches are denoted by $\bar{\mathbf{x}}_i$ for $i = 1, 2, \ldots, N - M$. For now, let f represent the function we wish to interpolate given the function values $f(\mathbf{x}_i) = f_i$ for each of the M patches in X.

Let $k(\mathbf{x}_i, \mathbf{x}_j)$ be a symmetric positive semi-definite kernel. We consider the following model for the RBF interpolant:

$$\tilde{f}(\mathbf{x}) = \sum_{j=1}^{M} \alpha_j k(\mathbf{x}, \mathbf{x}_j), \tag{6.1}$$

where $\mathbf{x}$ could be any patch, and α_j are coefficients to be determined. Evaluating (6.1) at the patches $\mathbf{x}_j$ for $j = 1, 2, \ldots M$ leads to the linear system of equations

$$\mathbf{f} = \mathbf{K}\boldsymbol{\alpha}, \tag{6.2}$$

where $\mathbf{f} = (f_1, \ldots, f_M)^T$, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)^T$, and $\mathbf{K}$ is M-by-M matrix with entries $\mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$.

Since $\mathbf{K}$ is symmetric, we can write $\mathbf{K} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$, where $\mathbf{U}$ is an orthogonal matrix and $\boldsymbol{\Lambda}$ is diagonal. That is, the kth diagonal entry of $\boldsymbol{\Lambda}$, denoted by $\tilde{\lambda}_k$, and the k^{th} column of $\mathbf{U}$, denoted by $\mathbf{u}_k$, satisfy $\mathbf{K}\mathbf{u}_k = \lambda_k \mathbf{u}_k$.

If we assume that $\mathbf{K}$ is nonsingular, from (6.2) we obtain

$$\boldsymbol{\alpha} = \mathbf{U}\boldsymbol{\Lambda}^{-1}\mathbf{U}^T\mathbf{f}.$$

Now, assume that $\mathbf{f} = \mathbf{u}_k$ for some $k \in \{1, 2, \ldots, M\}$. Under these assumptions, it is easy to see that $\boldsymbol{\alpha} = \tilde{\lambda}_k^{-1}\mathbf{u}_k$. Using this fact, each eigenvector of $\mathbf{K}$ can be extended via (6.1) to the function

$\tilde{\mathbf{u}}_k = (\tilde{u}_k(\bar{x}_1), \ldots, \tilde{u}_k(\bar{x}_{N-M}))^T$, defined on the $N - M$ points as

$$\tilde{\mathbf{u}}_k = \mathbf{B}^T(\tilde{\lambda}_k^{-1}\mathbf{u}_k), \tag{6.3}$$

where $\mathbf{B}$ is a M-by-$(N - M)$ matrix with entries $\mathbf{B}_{i,j} = k(\mathbf{x}_i, \bar{\mathbf{x}}_j)$. Equation (6.3) represents the RBF interpolant of the eigenvectors of the M-by-M matrix $\mathbf{K}$.

Because we are interested in computing the eigenvectors of the matrix $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$, we choose the RBF kernel to be

$$k(\mathbf{x}_i, \mathbf{x}_j) = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2)}{(d(\mathbf{x}_i)d(\mathbf{x}_j))^{1/2}},$$

where $d(\mathbf{x}_i) = \mathbf{D}_{ii}$ represents the ith diagonal entry of $\mathbf{D}$. It follows that if we obtain the eigenfunctions $\mathbf{u}_k$ defined on the subset of patches X by diagonalizing the M-by-M matrix $\mathbf{K}$, then, whenever $\lambda_k \neq 0$, equation (6.3) yields approximations to the eigenvector ϕ_k defined for all N patches in $\overline{X}$.

The authors [33] also propose that the approximation (6.3) actually diagonalize an approximation to a *dense* version of N-by-N weight matrix $\mathbf{W}$ defined in section 2.2. That is, the authors consider all possible weights between patches, not just those between ν nearest neighbors. Hence, let $\tilde{\mathbf{W}}$ represent the approximation to a dense $\mathbf{W}$. Associate the M points in the subset X with the first M rows of the matrix $\tilde{\mathbf{W}}$, and associate the $N-M$ points not in the subset with the remaining $N - M$ rows of $\tilde{\mathbf{W}}$. That is, row $1 \leq i \leq M$ of $\tilde{\mathbf{W}}$ corresponds to $\mathbf{x}_i$, while row $M < i \leq N$ corresponds to $\bar{\mathbf{x}}_{i-M}$. It follows that we can write the M extensions, given in (6.3), on all N patches as an N-by-M matrix

$$\tilde{\mathbf{U}} = \begin{pmatrix} \mathbf{U} \\ \mathbf{B}^T\mathbf{U}\mathbf{\Lambda}^{-1} \end{pmatrix}, \tag{6.4}$$

where the kth column of $\tilde{\mathbf{U}}$ equals $\tilde{\mathbf{u}}$ given in (6.3).

Instead of approximating the eigenvectors using columns of $\tilde{\mathbf{U}}$, we use columns of a different matrix $\tilde{\mathbf{V}}$, which "diagonalizes" the approximate $\tilde{\mathbf{W}}$. Specifically, as shown in [33], we can write $\tilde{\mathbf{W}} = \tilde{\mathbf{V}}\mathbf{\Sigma}\tilde{\mathbf{V}}^T$ for a N-by-M matrix $\tilde{\mathbf{V}}$ satisfying $\tilde{\mathbf{V}}^T\tilde{\mathbf{V}} = \mathbf{I}$ and a diagonal M-by-M matrix $\mathbf{\Sigma}$ if we

choose

$$\tilde{\mathbf{V}} = \begin{pmatrix} \mathbf{K} \\ \mathbf{B}^T \end{pmatrix} \mathbf{K}^{-1/2} \mathbf{\Psi} \mathbf{\Sigma}^{1/2}, \tag{6.5}$$

where diagonal $\mathbf{\Sigma}$ and M-by-M unitary $\mathbf{\Psi}$ are chosen to satisfy

$$\mathbf{\Psi} \mathbf{\Sigma} \mathbf{\Psi}^T = \mathbf{K} + \mathbf{K}^{-1/2} \mathbf{B} \mathbf{B}^T \mathbf{K}^{-1/2}.$$

It follows that the kth column of $\tilde{\mathbf{V}}$ is an alternative approximation to ϕ_k.

6.3.1 Choosing the subset of patches

6.3.1.1 Method One - Random Selection

In typical implementations of the Nyström extension, the M points in the subset are either fixed, or chosen with uniform probability from the original dataset [24, 33]. In the following, we initialize the subset X with a single patch that is selected with uniform probability from the N patches (i.e. with probability N^{-1}). For $k = 2, 3, \ldots, M$, the k^{th} point chosen to be in the subset is selected with probability $(N - (k-1))^{-1}$ from the $N - (k-1)$ points that do not already belong to the subset.

Selecting points in this way is highly efficient, but may betray the geometry of the data. For instance, a dataset of interest may contain points sampled from a low-density region of the $\mathbb{R}^d$. In this case, one would prefer having the points from low-density regions belong to the subset of M points, in order to better represent the true geometry of the original dataset using only the subset of M datapoints. This notion of preserving the original dataset's geometry as best as possible leads us to explore the following procedure for selecting the subset in a geometrically meaningful manner.

6.3.1.2 Method Two - Geometric Selection

Inspired by the coloring algorithm of AMG [15, 88], we say that patch $\mathbf{x}_i$ *strongly depends on* $\mathbf{x}_j$ if

$$\mathbf{W}_{i,j} \geq \alpha \max_{k \neq i} \mathbf{W}_{i,k}, \tag{6.6}$$

where $\alpha \in (0, 1)$. The condition (6.6) is satisfied if and only if

$$\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\max_{k \neq i} \|\mathbf{x}_i - \mathbf{x}_k\|^2} \leq \alpha.$$

The subset X is chosen to be a minimal set of points such that any $\mathbf{x}_i$ from the original dataset strongly depends on at least one point $\mathbf{x}_j$ in the subset. The idea is that relatively small distances between $\mathbf{x}_i$ and some of its ν-nearest-neighbors can be neglected. This criterion ensures that all points x_i have a relatively close neighbor that is part of the subset of patches, X, used to interpolate from.

In effect, subsampling a set of patches in this way adds patches that to the subset X if they are located in a region of $\mathbb{R}^d$ that has a relatively low density of patches because in such a region, this patch has no other neighbors that it is strongly connected to. Thus, it must be included in the subset X.

6.3.2 Numerical experiments

In this section, we investigate the performance of the extensions (6.4) and (6.5) using two patch sets: one associated with the signal $x(t) = \sin(2\pi t)$, and another associated with the clown image in Figure 6.2. We will call these signals the circle dataset and clown dataset, respectively. For the circle, we extract patches using $d = 2$ and $\Delta t = 4^{-1}$, in order to produce a perfect circle in phase space. The circle dataset exemplifies an ideal dataset, where points are equally spaced on the underlying geometry. Note that there are 2^{10} points in the circle dataset. For the clown data, we extract 5-by-5 pixel patches, maximally overlapping in the image plane, leading to 2^{12} patches.

6.3.2.1 Computing the weight matrix

Given a set of patches $\{\mathbf{x}_i\}_{i=1}^N$, entries of the matrix $\mathbf{W}$ are computed in this section as follows. First, the $\nu = 2\log(N)$ nearest neighbors are computed for each $\mathbf{x}_i$ and the distances between $\mathbf{x}_i$ and its ν nearest neighbors are recorded. The minimum and maximum distances between $\mathbf{x}_i$ and

its ν nearest neighbors are used to define the kernel's scale parameter σ^{-2} as

$$\sigma^{-2} = \frac{\log \max \|\mathbf{x}_i - \mathbf{x}_j\|^2 - \log \min \|\mathbf{x}_i - \mathbf{x}_j\|^2}{\max \|\mathbf{x}_i - \mathbf{x}_j\|^2 - \min \|\mathbf{x}_i - \mathbf{x}_j\|^2}. \tag{6.7}$$

Using this value of σ^{-2}, we set $(\mathbf{W})_{i,j} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2)$ if $\mathbf{x}_i$ is one of the ν nearest neighbors of $\mathbf{x}_j$ or if $\mathbf{x}_j$ is one of the ν nearest neighbors of $\mathbf{x}_i$, otherwise, we set $\mathbf{W}_{i,j} = 0$. For a fixed ν, choosing σ^{-2} in this way maximizes

$$f(\sigma^{-2}) = \max_{1 \le i,j \le N} \mathbf{W}_{i,j} - \min_{1 \le i,j \le N} \mathbf{W}_{i,j}.$$

In other words, choosing σ as in (6.7) will maximize the range of weights that we observe in the weight matrix.

6.3.2.2 Comparing the methods for selecting the subsets to interpolate from

For both the circle and the clown datasets, we compute the random and geometric subsets of patches X, as described in section 6.3.1. For the circle data, we set $\alpha = 0.9$. For the roof and clown data, we set $\alpha = 0.1$. These choices of α lead to subsets X that contain roughly 10% of the original set of patches $\overline{X}$.

The two subsets X that we will interpolate from are shown for the circle and clown datasets in Figures 6.1 and 6.2, respectively. Notice that geometric selection of the subset leads to nearly uniformly separated patches in the circle dataset and patches in the clown dataset that are more clustered around details in the image. Also notice that there are relatively little patches in the subset that are extracted from the same smooth region of the image. We understand this as follows: The smooth content of the image puts patches very close together in $\mathbb{R}^d$, and when a patch containing this smooth content is chosen in the subset using the geometric selection of section (6.3.1.2), the other patches that are close in $\mathbb{R}^d$ and strongly depend on $\mathbf{x}_n$ do not need to be included in subset X.

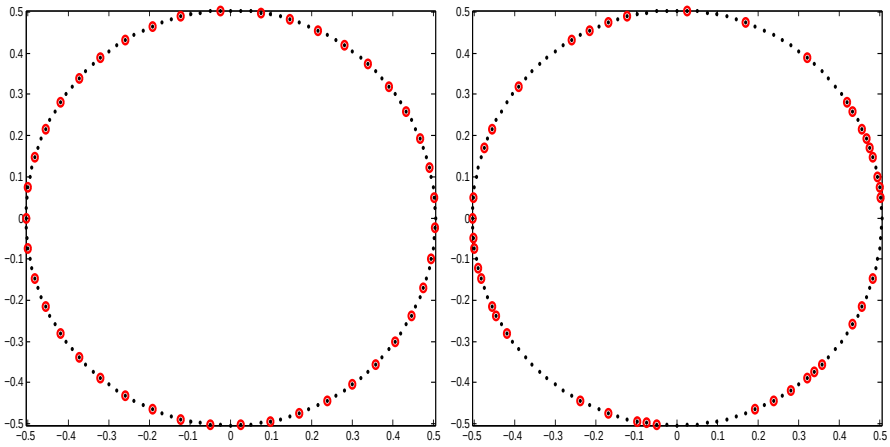

Figure 6.1: The original circle dataset and each subsample: random subsampling on the right, and geometric subsampling based on AMG coloring on the left.

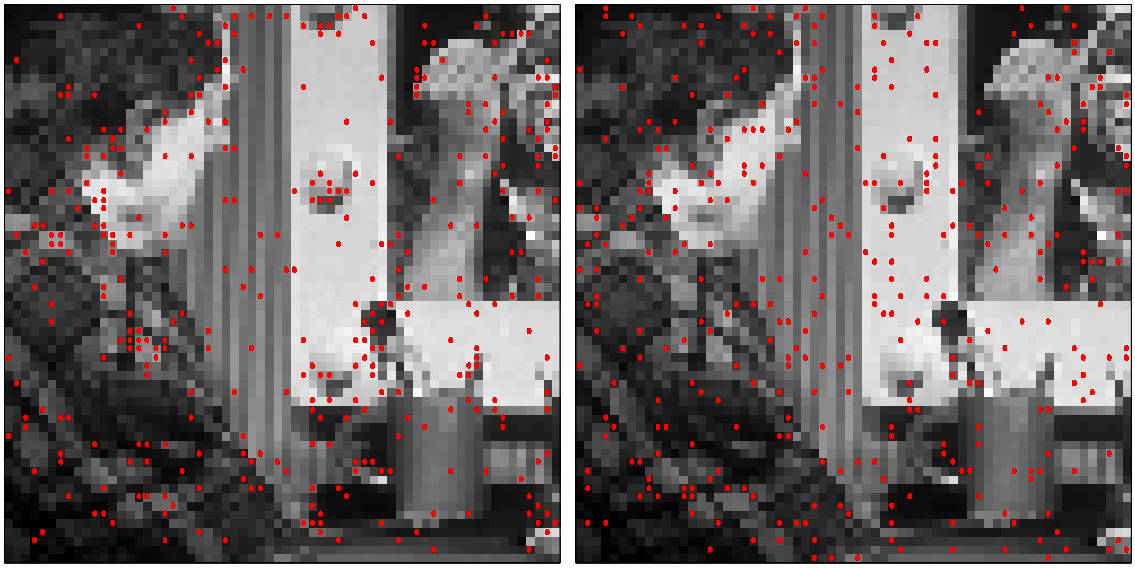

Figure 6.2: Subsets of patches extracted from clown dataset. Right: Random selection of the subset. Left: Geometric selection of the subset.

6.3.2.3 Comparing approximate eigenfunctions

Figure 6.3 shows the extension (6.4) that is obtained by simply interpolating, and the extension (6.5) that is obtained by enforcing the diagonalization constraint. When enforcing the diagonalization constraint, we see increased accuracy. In addition, we see that the subset of patches X that

is chosen using geometric selection visually leads to much more accurate approximations. Indeed, in all our experiments, the random subset of patches leads to the least accurate approximations. For this reason, we abandon choosing the subset of patches at random.

The times required to compute the eigenfunctions as a function of the subset size is given in Table 6.1. To compute the same number of eigenvectors using the entire set of N patches requires 0.45 seconds for the circle data, and 51.0 seconds for the clown data. In addition, the error, as measured using the 2-norm is also provided. The error between the extended eigenvectors and those computed using all N patches is near machine precision for the circle dataset. For the clown dataset, the error is 0.0013. Therefore, although we can reduce the time it takes to compute the eigenvectors associated with the clown dataset by a factor of 10, the relative error is increased by roughly a factor of 10.

In Figure 6.4, we see that the subset of patches chosen using geometric selection always produce more accurate eigenvectors, as measured by the Euclidean norm of the residual, given by

$$\|(\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} - \lambda_k\mathbf{I})\tilde{\mathbf{v}}_k\|_2,$$

where $\tilde{\mathbf{v}}_k$ represents the approximate eigenvector, and the matrices $\mathbf{D} \in \mathbb{R}^{N\times N}$, $\mathbf{W} \in \mathbb{R}^{N\times N}$ and eigenvalue λ_k are computed using all N patches in $\overline{X}$.

In Figure 6.5, we demonstrate how the accuracy of the approximate eigenvectors depends on the size of the subset X. Unsurprisingly, as M increases, the approximate eigenvectors become more accurate. As before, using the approximation (6.5), which enforces the diagonalization constraint, leads to more accurate approximations.

Table 6.1: Relative error in Nyström extension associated with circle dataset ($N = 2^{10}$ patches), and time required to compute. Time required to compute eigenvectors using N patches is 0.45 seconds.

M	Relative Error	Time Required (s)
50	6.97×10^{-2}	0.03
150	2.94×10^{-5}	0.27
250	2.24×10^{-10}	0.71
350	3.29×10^{-10}	1.70
450	4.82×10^{-11}	3.54

Table 6.2: Relative error in Nyström extension associated with clown dataset ($N = 2^{12}$ patches), and time required to compute. Time required to compute eigenvectors using N patches is 51.0 seconds.

M	Relative Error	Time Required (s)
81	0.0347	0.21
409	0.0325	4.3
1024	0.0311	86

6.4 The multi-level option

The out-of-sample extensions described above can be interpreted as prolongation in the context of multigrid solvers [88]. This realization leads us to consider using multilevel techniques to efficiently solve for the eigenpairs (λ_k, ϕ_k) when N is large. In this section, we report experiments based on an implementation of an AMG-based eigensolver that is described in [52] in order to obtain approximations to (λ_k, ϕ_k). This eigensolver iteratively refines its approximation to the first $K \geq 1$ eigenvectors ϕ_k.

Figure 6.6 shows the 2-norm of the residual for approximations of the first seven modes obtained via the AMG-based eigensolver and a standard Lanczos eigensolver. We see that the residual error in the AMG-based approximations associated with the circle data are nearly at machine precision after 13 iterations. Figure 6.7 shows the evolution of the residual error as a function of iteration. It is clear that just a few more iterations of the AMG V-cycle applied to the

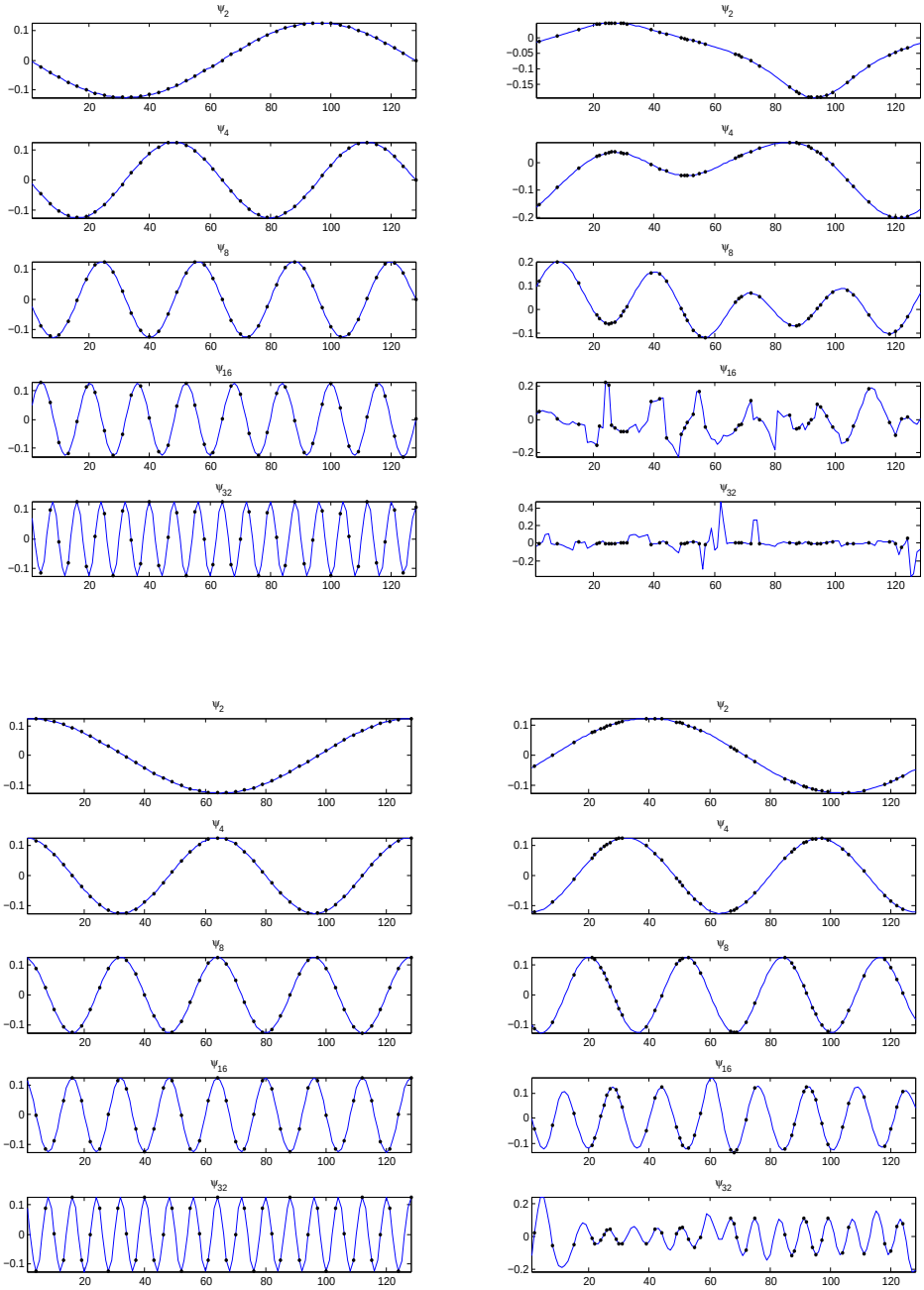

Figure 6.3: A subset of the extensions (6.4) and (6.5). Patches in the subset are indicated by black dots overlaid on the eigenfunctions. Left: subset selected geometrically. Right: subset selected randomly.

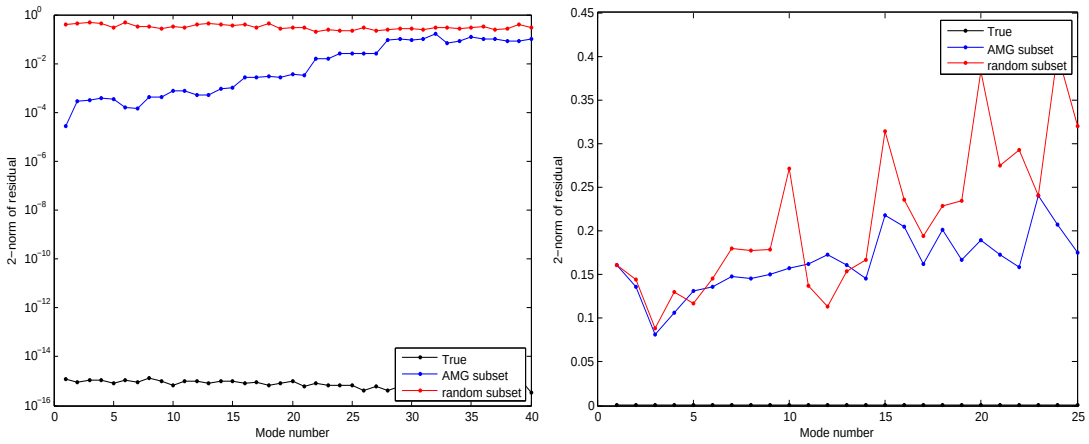

Figure 6.4: The quality of the approximations of the set of eigenvectors as measured using the 2-norm of the residual. The black curve represent the residual error in the eigenvectors that are computed using all N patches. Left: error in approximations associated with circle data. Right: error associated with clown data.

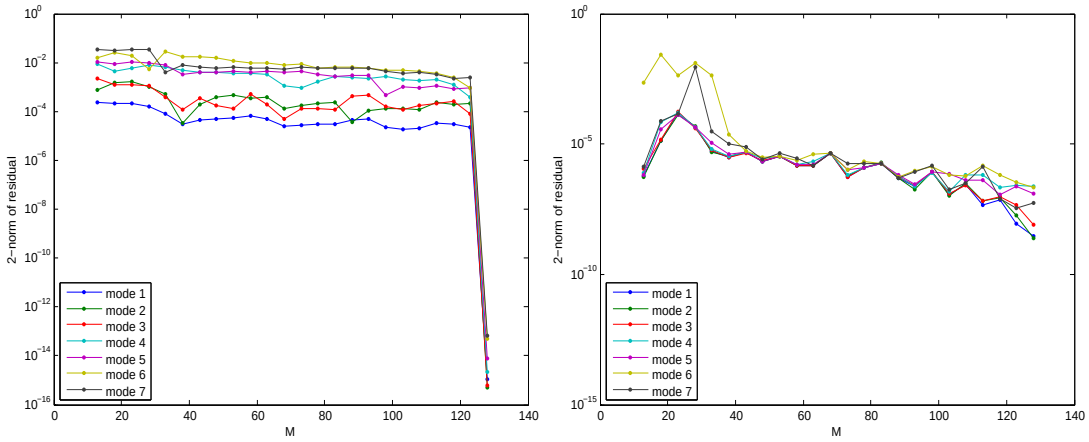

Figure 6.5: Residual errors when extending modes $\mathbf{u}_1, \ldots, \mathbf{u}_7$ from M datapoints to the N datapoints. Left: extension given by (6.4), without enforcing diagonalization constraint. Right: extension given by (6.5), which enforces the diagonalization constraint. The subset of M patches is chosen using geometric sampling. We choose shape parameter $\gamma = 1$ to produce the lowest errors out of possible parameters in the set $\{0.1, 1, 10, 100\}$.

circle dataset will cause all seven approximations to reach machine precision, provided the convergence rates do not change.

With the image patch datasets, the residual error does not reach machine precision, nor does it decay as rapidly with each iteration. Figure 6.7 shows that all but the seventh mode of the clown

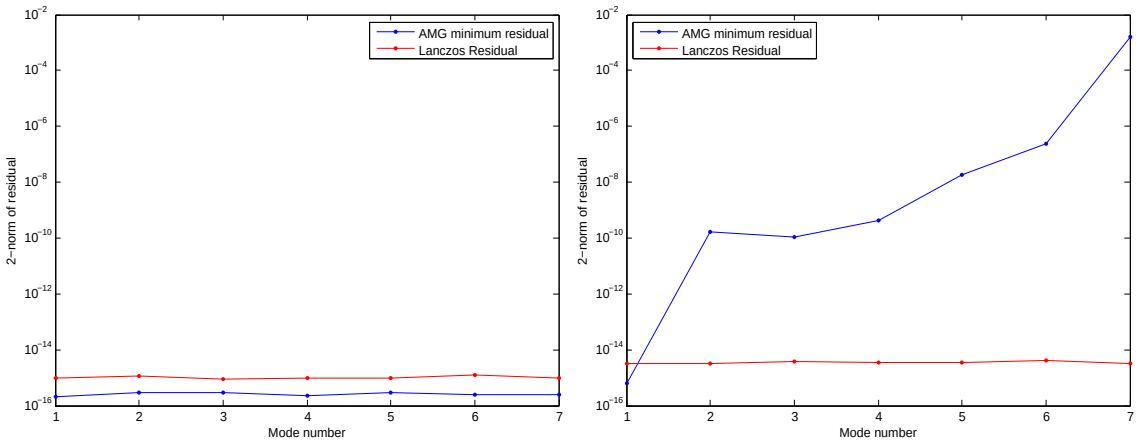

Figure 6.6: Residual errors and angles between approximations. Top row corresponds to the circle dataset. Middle row corresponds to the roof dataset. Bottom row corresponds to the clown dataset.

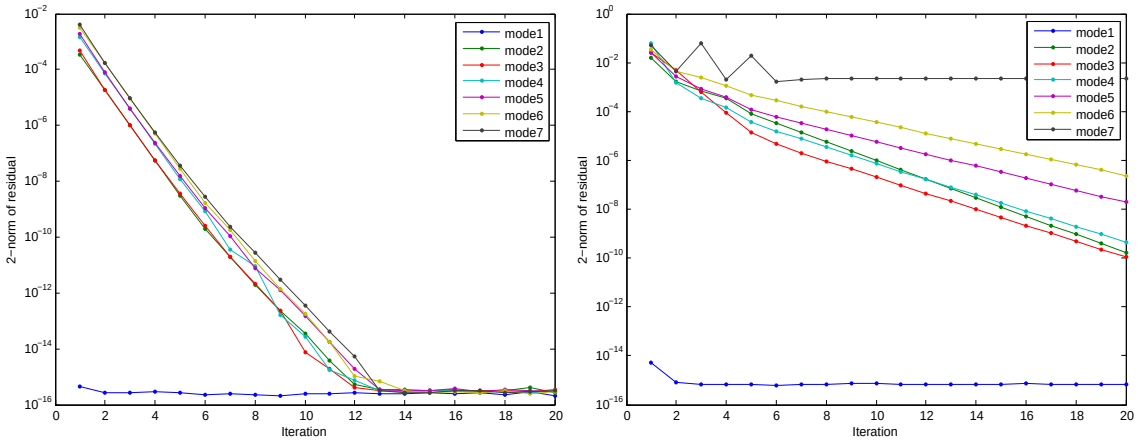

Figure 6.7: The 2-norm of the residual error as a function of iteration count. Left: circle dataset. Right: clown dataset.

dataset seems to converge.

6.5 Conclusion

Approximating the eigenvectors ϕ_k using the Nyström extension or using multilevel iterations based on algebraic multigrid avoids the computational cost of typical eigenvalue problems, which may require $\mathcal{O}(N^2)$ or $\mathcal{O}(N^3)$ computations, depending on which subspaces of the eigenspace are

computed, and the sparsity of the matrix $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$.

We have demonstrated that the Nyström extension is more accurate when choosing the subset of patches to interpolate from in a way that is aware of the underlying geometry. We also demonstrated that the additional diagonalization constraint, presented in [33], also leads to a more accurate extension, but at the cost of diagonalizing an M-by-M matrix.

Finally, despite the success of the eigensolver on simple datasets, presented in [52], we demonstrated that the complexity of the geometry in the patch-set slows convergence of the approximate eigenvectors to the true eigenvectors. This slow convergence may be avoided if Gauss-Seidel relaxation were swapped with Kaczmarz relaxation inside each iteration.

Chapter 7

Conclusion

In this thesis, we proved theoretical results that characterize the geometry of a signal or image's patch-set as points in Euclidean space, and as vertices of a graph. Our results establish that a parametrization of a signal's patch-set based on a random walk's commute time between vertices of the associated patch-graph is able to partition a signal's patch-set by relatively concentrating patches that exhibit rapid change, are well-separated, and inefficiently represented as points in $\mathbb{R}^d$. In addition, our experiments with real seismic data are encouraging, and suggest that our theoretical conclusions of the previous chapters account for the success of diffusion-based parametrizations of signal or image patch-sets.

We discuss parameter selection in section 7.1. In section 7.2, we point out some possible extensions of our approach. In section 7.3, we discuss related work and connect our interpretation with the diffusion interpretation of [80]. Finally, we discuss open questions in section 7.4.

7.1 Guides for selecting parameters

7.1.1 Choosing the patch size

In this work we are interested in the *local* behavior of the image, and therefore d should remain of the order of what we consider to be the local scale. We also note that as d becomes large, the number of available patches (N/d) becomes smaller, making the estimation of the geometry of the patch-set more difficult, since patches now live in a high-dimensional space. Another consequence of the "curse of dimensionality" is that the distance between patches becomes less informative for

large values of d. If the original signal is oversampled with respect to the true physical processes at stake, then one can coarsen the sampling of the patch-set in the signal domain. In practice, it would be more advisable to coarsen the underlying continuous patch-set, which is a nontrivial question.

7.1.2 Choosing edge weights

In general, two principles guide the choice of edge weights in the patch-graph. On the one hand, patches that are very close should be connected with a large weight (short distance), while patches that are faraway should have a very small weight along their mutual edge. This principle is equivalent to the idea of only trusting local distances in $\mathbb{R}^d$. Such a requirement is intuitively reasonable if we assume that the patch-set represents a discretization of a nonlinear manifold in $\mathbb{R}^d$. In this situation, we know that when the points on the manifold are very close to another, the *geodesic distance* is well approximated by the Euclidean distance. Conversely, because of the presence of curvature, the Euclidean distance is a poor approximation to the geodesic distance on the manifold when points are far apart. Because the only information available to us is the Euclidean distance between patches, we should not trust large Euclidean distances.

On the other hand, as observed in Section 3.7, the fast patches, which contain rapid changes, are all very far apart (large $\rho^2(\mathbf{x}_n, \mathbf{x}_m)$). Therefore the probability that the random walk escapes the fast patch $\mathbf{x}_n$ and jumps to a different patch $\mathbf{x}_m$, which is given by

$$\frac{w_{n,m}}{\sum_l w_{n,l}} = \frac{e^{-\rho^2(\mathbf{x}_n, \mathbf{x}_m)/\sigma^2}}{\sum_l w_{n,l}},$$

is always much smaller than the probability of staying at $\mathbf{x}_n$, which is given by

$$\frac{1}{\sum_l w_{n,l}}.$$

In order to avoid trapping the random walk at a fast patch, we "saturate" the distance function by choosing σ to be very large. In this case, for all the nearest neighbors $\mathbf{x}_m$ of $\mathbf{x}_n$, we have $w_{n,m} \approx 1$, and the transition probability is the same for all the neighbors, $\mathbf{P}_{n,m} \approx 1/\nu$. This

choice of σ promotes a very fast diffusion of the random walk locally on the patch-graph. We note that choosing a large σ may be avoided if self-connections are not enforced (i.e. $w_{n,n} = 0$). However, self-connections are a necessary technical requirement to prove that the Markov process is aperiodic, which is required to prove the equality (4.4) [23].

We note that choosing σ to be very large does not entirely obliterate the information provided by the mutual distance between patches. Indeed, a metric ρ is used to select the nearest neighbors of each patch, and therefore allows us to define a notion of a local neighborhood around each patch. Choosing σ to be very large forces a very fast diffusion within this neighborhood, irrespective of the actual distances ρ. Alternatively, we could consider choosing σ to vary adaptively from one neighborhood to another. The parameter σ could be small when patches are extremely close to one another, while σ could be large when the patches are at a large mutual distance of one another. This notion is the foundation of the self-tuning weight matrix, which adjusts its weights based on a point's local neighborhood [59].

7.2 Extensions and generalizations

In general, the patch-set of an image consists of more than two homogeneous subsets. For example, one could partition an image patch-set into uniform patches, edge patches, and texture patches. Our experience with a generalization of the time-frequency signal model from section 4.5 indicates that we can still separate the patches when the signal is composed of up to four different local behaviors that are specified by four different values of the parameter in the autocorrelation function (see Figure 7.1).

Finally, although we consider a graph whose vertices are patches from a time series or image, the core of our assumptions is about the structure of the graph itself — the coherence between "slow vertices" of the graph may be a consequence of something more general than simply temporal or spatial coherence in the time series or image. For example, if vertices of a graph represent sites on the internet and edges exist between websites with similar content, then the coherence in the daily news across different reporting websites may lead to a geometry that is similar to the slow

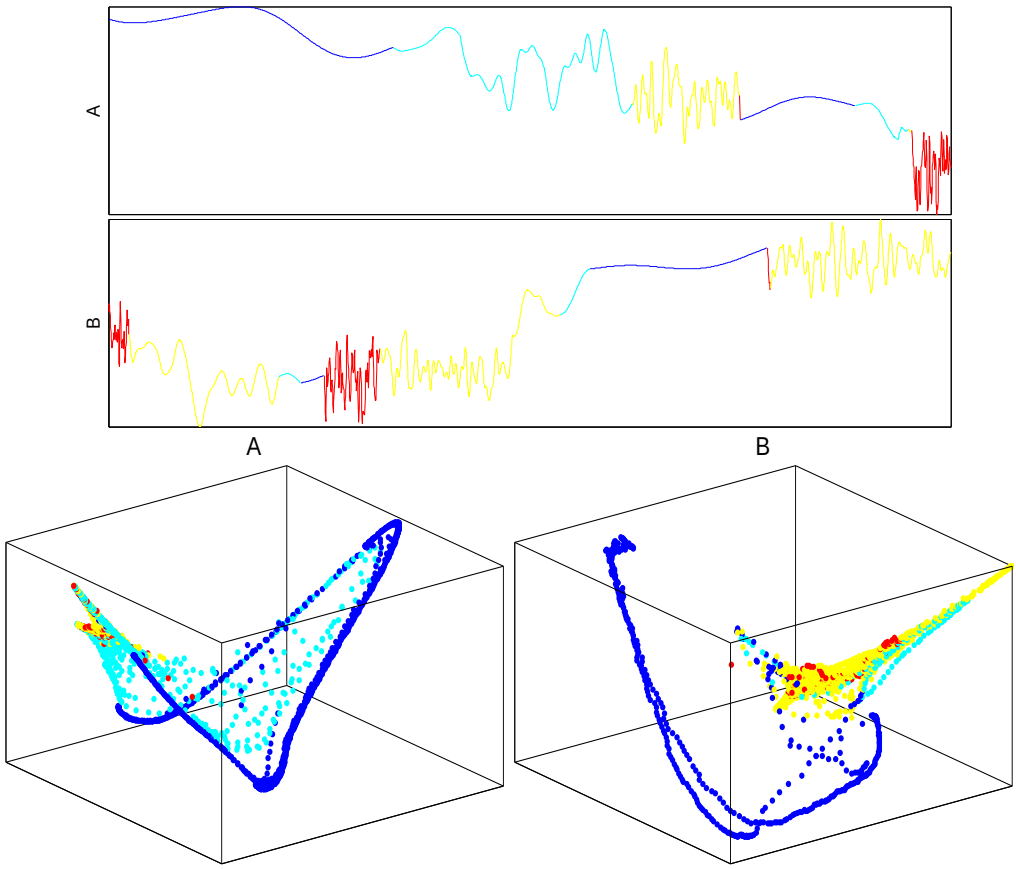

Figure 7.1: The top two plots show two realizations from a generalization of the time-frequency signal model of section 4.5. The four colors correspond to four covariance parameters, which creates fast patches with various degrees of local change. We see that the "fastest" patches (orange and red) are those that are most concentrated in the bottom two plots, which show the patch-set mapped through Φ in (4.8), using $d' = 3$.

graph's geometry. Our theoretical conclusions, therefore, can be extended to the parametrization of any graph whose geometry is similar to the assumed form of the patch-graph's geometry, regardless of where it originates.

7.3 Related work

Diffusion on the patch-graph has proven useful in texture analysis/synthesis [56], multi-modal image registration [92], super-resolution [71], and denoising [16, 81, 83]. These works argue that the graph-based perspective is useful based on numerical experiments using several images. In addition,

the works [40, 13, 98, 79, 60, 53] also provide empirical evidence that a graph-representation of a time series can be useful for understanding the underlying dynamical system which generated the data.

We go beyond providing empirical evidence, and theoretically justify the utility in a graph-based approach for detecting rapid change in the underlying signal. We note that the work [80] also offers a theoretical explanation to the success of a graph-based approach for removing noise from a signal, and so it is interesting to contrast their work with ours. As described in [80], Singer et al., treat the matrix $\mathbf{P}$ as a filter, which acts on an N-dimensional column-vector-representation of the signal. The matrix-vector multiply is regarded as evolving the diffusion process on the patch-graph for a time-step on the order of σ. Their results rely on the convergence of a scaled $\mathbf{P}$ to the backward Fokker-Planck operator, and the special form of this operator and its eigenfunctions when the signal is either a one-dimensional constant perturbed by Gaussian noise, or a one-dimensional step function also contaminated by Gaussian noise.

Besides the fact that we consider a class of signals that is more diverse than one containing noisy constants or step functions, the main difference between our analysis and [80] is that the convergence of $\mathbf{P}$ to the backward Fokker-Planck operator relies on both $N \to \infty$ and $\sigma \to 0$, while our results rely on a ν nearest neighbor graph and large σ (See section 7.1) and requiring just $N \to \infty$. We also note that Singer et al. study the *mean first-passage time* between patches extracted from the noisy step function. The mean first-passage time is derived from a diffusion that is intimately related to the random walk, which is used to define the commute time. Singer et al. explain the existence of a large mean first-passage time between patches extracted from either side of the step function's discontinuity using an energy argument. In particular, they argue that a high density of patches is associated with a lower potential energy, and, consequently, it will take longer for a random process to exit the well with such a low potential. Finally, because our assumptions are more qualitative than technical, our results are not limited to patches of size $d = 1$, as are the results in [80].

The energy argument in [80] adds an interesting interpretation to our analysis. Following the

energy perspective, the slow patches can be thought of as points sampled from a probability density function P defined on $\mathbb{R}^d$ that has most of its mass localized in a small region. This localization leads to a potential $U = -\log P$ with a deep, narrow well that would be difficult for a process to exit from. In some sense, this argument agrees with our findings that the average commute time between slow patches is very large, and thus, the random walker would spend relatively more time in portions of the patch-graph corresponding to the slow patches before jumping to a patch that is temporally far away.

From a more general perspective, this work presents an investigation into the diffusion process on the different models graphs from section 4.4.2. The works [68, 85, 10, 63] also focus on characterizing graphs or networks using a diffusion process. These works are motivated by physical problems such as transport in disordered media, neuron firing, or energy flow on power-grids instead of applications in signal analysis. Nevertheless, these works also characterize graphs using the (mean) first-passage time.

7.4 Open questions

First, although we obtained estimates for average commute times in the fast and slow graph models considered separately, it is desirable to obtain similar estimates when each is considered part of the fused graph model. In particular, it would be interesting to understand how the commute time between vertices coming from *distinct* components of the fused graph model would behave as a function of N and L. Indeed, Figure 4.6 suggests that there are likely extensions of our results to the commute times on the fused graph. Also, it would be interesting to understand these results as a function of the separation between the components, q.

Finally, we point out that the upper bound on $\kappa_{\mathcal{F}}$ is increasing with L even though we expect that the commute time between points of $\mathcal{F}(N, p)$ would decrease as L increases. The reason for this apparent inconsistency is that the proof of (4.16) relies on the fact that the effective resistance between the two terminals of the associated electrical circuit is bounded from above by their geodesic distance on the graph [17]. A more effective inequality that could improve the upper

bound (4.16) relies on knowing the number of paths s of length at most l between the terminal nodes of the circuit. If we knew this distribution, then we could use the fact that the commute time is bounded from above by a constant times the ratio l/s [17], which would decrease the upper bound in (4.16).

Bibliography

[1] H. Abarbanel, R. Brown, J. Sidorowich, and L. Tsimring. The analysis of observed chaotic data in physical systems. Rev. Modern Phys., 65:1331–1392, 1993.

[2] P. Abry and F. Sellan. The wavelet-based synthesis for fractional Brownian motion proposed by F. Sellan and Y. Meyer : Remarks and fast implementation. Applied and Computational Harmonic Analysis, 3:377–383, 1996.

[3] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. Signal Processing, IEEE Transactions on, 54(11):4311 –4322, Nov. 2006.

[4] R. Allen. Automatic phase pickers: Their present use and future prospects. Bull. Seism. Soc. Am., 68:1521–1532, 1982.

[5] K. Anant and F. Dowla. Wavelet transform methods for phase identification in three-component seismograms. Bull. Seism. Soc. Am., 87:1598, 1997.

[6] T. Bardainne, P. Gaillot, N. Dubos-Sallée, J. Blanco, and G. Sénéchal. Characterization of seismic waveforms and classification of seismic events using chirplet atomic decomposition. Example from the Lacq gas field (Western Pyrenees, France). Geophysical Journal International, 166(2):699–718, 2006.

[7] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computations, 15:1373–1396, 2003.

[8] Y. Ben-Zion. Collective behavior of earthquakes and faults: Continuum-discrete transitions, progressive evolutionary changes, and different dynamic regimes. Rev. Geophys., 46, 2008.

[9] O. Bénichou, C. Chevalier, J. Klafter, B. Meyer, and R. Voituriez. Geometry-controlled kinetics. Nature Chemistry, 2(6):472–477, 2010.

[10] O. Bénichou and R. Voituriez. Narrow-escape time problem: Time needed for a particle to exit a confining domain through a small window. Phys. Rev. Lett., 100(16):168105, Apr 2008.

[11] P. Bérard, G. Besson, and S. Gallot. Embeddings Riemannian manifolds by their heat kernel. Geometric and Functional Analysis, 4(4):373–398, 1994.

[12] J. Berger and R. Sax. Seismic detectors: the state of the art. Technical report, VELA Seismological Center, Alexandria, VA., 2001.

[13] E. P. Borges, D. O. Cajueiro, and F. S. Andrade. Mapping dynamical systems onto complex networks. The European Physical Journal B - Condensed Matter and Complex Systems, 58:469–474, 2007.

[14] P. Bremaud. Markov Chains. Springer Verlag, 1999.

[15] M. Brezina, R. Falgout, S. MacLachlanT. Manteuffel, S. McCormick, and J. Ruge. Adaptive algebraic multigrid. SIAM J. Sci. Comput., 27(4):1261–1286, 2006.

[16] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. Multiscale Modeling and Simulation, 4:490–530, 2005.

[17] A. K. Chandra, P. Raghavan, W. L. Ruzzo, and R. Smolensky. The electrical resistance of a graph captures its commute and cover times. In Proceedings of the twenty-first annual ACM symposium on Theory of computing, STOC '89, pages 574–586, New York, NY, USA, 1989. ACM.

[18] B. Chapelle, O. Schölkopf and A. Zien, editors. Semi-Supervised Learning. MIT Press, Cambridge, MA, 2006.

[19] B. Chouet and H. Shaw. Fractal properties of tremor and gas piston events observed at Kilauea Volcano Hawaii. J. Geophys. Rev, 96:10177–10189, 1991.

[20] F. Chung. Spectral Graph Theory. American Mathematical Society, 1997.

[21] F. Chung, L. Lu Linyuan, and V. Vu. Spectra of random graphs with given expected degrees. Proceedings of the National Academy of Sciences, 100(11):6313–6318, 2003.

[22] A. Cohen and J. P. D'Ales. Nonlinear approximation of random functions. SIAM J. Appl. Math., 57:518–540, April 1997.

[23] R. R. Coifman and S. Lafon. Diffusion maps. Applied and Computational Harmonic Analysis, 21:5–30, 2006.

[24] Ronald R. Coifman and Stphane Lafon. Geometric harmonics: A novel tool for multiscale out-of-sample extension of empirical functions. Applied and Computational Harmonic Analysis, 21(1):31 – 52, 2006. Diffusion Maps and Wavelets.

[25] S. Condamin, O. Bénichou, V. Tejedor, R. Voituriez, and J. Klafter. First-passage times in complex scale-invariant media. Nature, 450(7166):77–80, 2007.

[26] E. De Lauro, E. De Martino, S. Del Pezzo, M. Falanga, M. Palo, and R. Scarpa. Model for high-frequency strombolian tremor inferred by wavefield decomposition and reconstruction of asymptotic dynamics. J. Geophys. Res., 113:B02302, 2008.

[27] S. De Martino, M. Falanga, and C. Godano. Dynamical similarity of explosions at stromboli volcano. Geophys. J. Intern., 157:1247–1254, 2004.

[28] R. Di Stefano, F. Aldersons, E. Kissling, P. Baccheschi, and C. Chiarabba. Automatic seismic phase picking and consistent observation error assessment: application to the italian seismicity. Geophys. J. Intern., 165:121–134, 2006.

[29] P. G. Doyle and J. L. Snell. Random Walks and Electric Networks. ArXiv Mathematics e-prints, 2000.

[30] R. Durrett. Random Graph Dynamics. Cambridge, 2007.

[31] J. P. Eckmann, S. O. Kamphorst, and D. Ruelle. Recurrence plots of dynamical systems. Europhysics Letters, 4:973–977, 1987.

[32] P. Erdos and A. Renyi. On the evolution of random graphs. Publ. Math. Inst. Hung. Acad. Sci, 5:17–61, 1960.

[33] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the Nyström method. IEEE Trans. Pattern Anal. Mach. Intell., 26(2):214–225, 2004.

[34] V. Frede and P. Mazzega. Detectability of deterministic non-linear processes in earth rotation time-series embedding. Geophys. J. Intern., 137:551–564, 1999.

[35] V. Frede and P. Mazzega. Detectability of deterministic non-linear processes in earth rotation time-series embedding. Geophys. J. Intern., 137:567–579, 1999.

[36] H. Freedman. "The little variable factor." A statistical discussion of the reading of seismograms. Bull. Seism. Soc. Am., 56:593–604, 1966.

[37] W. Freiberger. An approximate method in signal detection. Quarterly App. Math, 20:373–378, 1963.

[38] A. Fronczak, P. Fronczak, and J. A. Hołyst. Average path length in random networks. Phys. Rev. E, 70(5), Nov 2004.

[39] J. Galiana-Merino, J. Rosa-Herranz, and S. Parolai. Seismic p phase picking using a kurtosis-based criterion in the stationary wavelet domain. IEEE Trans. Geosci. Remote Sens., 46:3815–3826, 2002.

[40] Z. Gao and N. Jin. Complex network from time series based on phase space reconstruction. Chaos: An Interdisciplinary Journal of Nonlinear Science, 19(3):033137, 2009.

[41] J. B. Garnett. Bounded Analytic Functions. Springer-Verlag, 1st edition, 2007.

[42] P. Gendron, J. Ebel, and D. Manolakis. Rapid joint detection and classification with wavelet bases via bayes theorem. Bull. Seism. Soc. Am., 90:764, 2000.

[43] R. Gilmore. Topological analysis of chaotic dynamical systems. Rev. Modern Phys., 70:1455–1529, 2000.

[44] C. Godano, C. Cardaci, and E. Privitera. Intermittent behaviour of volcanic tremor at Mt. Etna. Pure appl. Geophys., 147:729–744, 1996.

[45] L. Grafakos. Modern Fourier Analysis. Springer-Verlag, 2nd edition, 2008.

[46] P. Grassberger and I. Procaccia. Measuring the strangeness of strange attractors. Physica D, 9(1-2):189–208, October 1983.

[47] T. Hastie, R. Tibshinari, and J. Freedman. The elements of statistical learning. Springer Verlag, 2009.

[48] P. Indyk. Nearest neighbors in high-dimensional spaces. In Jacob E. Goodman and Joseph O'Rourke, editors, Handbook of Discrete and Computational Geometry, chapter 39. CRC Press, 2004. 2rd edition.

[49] K. Konstantinou. Deterministic non-linear source processes of volcanic tremor signals accompanying the 1996 Vatnajokull eruption, central Iceland. Geophys. J. Intern., 148:663–675, 2002.

[50] K. Konstantinou and V. Schlindwein. Nature, wavefield properties and source mechanism of volcanic tremor: a review. J. Volcanol. Geoth. Res., 119:663–675, 2002.

[51] L. Küperkoch, T. Meier, J. Lee, and W. Friederich. Automated determination of p-phase arrival times at regional and local distances using higher order statistics. Geophys. J. Intern., 181:1159–1170, 2010.

[52] Dan Kushnir, Meirav Galun, and Achi Brandt. Efficient multilevel eigensolvers with applications to data analysis tasks. IEEE Trans. Pattern Anal. Mach. Intell., 32(8):1377–1391, 2010.

[53] L. Lacasa, B. Luque, F. Ballesteros, J. Luque, and J. Nu no. From time series to complex networks: The visibility graph. Proceedings of the National Academy of Sciences, 105(13):4972–4975, April 2008.

[54] A. B. Lee, K. S. Pedersen, and D. Mumford. The nonlinear statistics of high-contrast patches in natural images. IJCV, 54(1-3):83–103, August 2003.

[55] L. Lovász. Random walks on graphs: A survey. In Combinatorics: Paul Erdös is eighty. János Bolyai Math. Soc, 1993.

[56] J. Lu, J. Dorsey, and H. Rushmeier. Dominant texture and diffusion distance manifolds. Computer Graphics Forum, 28(2):667–676, 2009.

[57] R. Lyons and Y. Peres. Probability on trees and networks. In preparation. Available at http://mypage.iu.edu/ rdlyons.

[58] S. Mallat. A Wavelet Tour of Signal Processing. Academic Press, 1999.

[59] L. Z. Manor and P. Perona. Self-tuning spectral clustering. In Proceedings of the 18th Annual Conference on Neural Information Processing Systems (NIPS'04), 2004.

[60] N. Marwan, J. F. Donges, Y. Zou, R. V. Donner, and J. Kurths. Complex network approach for recurrence analysis of time series. Physics Letters A, 373(46):4246 – 4254, 2009.

[61] N. Marwan, M. Carmen Romano, M. Thiel, and J. Kurths. Recurrence plots for the analysis of complex systems. Physics Reports, 438(5-6):237–329, 2007.

[62] Jiri Matousek. Lectures on Discrete Geometry. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[63] M. Moreau, O. Bénichou, C. Loverdo, and R. Voituriez. Chance and strategy in search processes. Journal of Statistical Mechanics: Theory and Experiment, 2009(12):P12006, 2009.

[64] D. Mount and S. Arya. Approximate nearest neighbors library, 2006. Software available at http://www.cs.umd.edu/ mount/ANN/ Accessed December 2009.

[65] S. Nippress, A. Rietbrock, and A. Heath. Optimized automatic pickers: application to the ancorp data set. Geophys. J. Intern., 181:911–925, 2010.

[66] B. A. Olshausen and D. J. Field. Natural image statistics and efficient coding. In Network: Computation in Neural Systems, 7:333–339, pages 333–339, 1996.

[67] C. Panagiotakis, E. Kokinou, and F. Vallianatos. Automatic p-phase picking based on local-maxima distribution. IEEE Trans. Geosci. Remote Sens., 46:2280–2287, 2008.

[68] P. E. Parris and V. M. Kenkre. Traversal times for random walks on small-world networks. Phys. Rev. E, 72(5):056119, Nov 2005.

[69] K. S. Pedersen and A. B. Lee. Toward a full probability model of edges in natural images. In in Proc. of ECCV02, pages 328–342. Springer Verlag, 2002.

[70] L. Persson. Statistical tests for regional seismic phase characterizations. J. Seismol., 7:19–33, 2003.

[71] M. Protter, M. Elad, H. Takeda, and P. Milanfar. Generalizing the nonlocal-means to super-resolution reconstruction. Image Processing, IEEE Transactions on, 18(1):36 –51, 2009.

[72] C. Richter. An instrumental earthquake magnitude scale. Bull. Seism. Soc. Am., 25, 1935.

[73] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. SCIENCE, 290:2323–2326, 2000.

[74] C. Saragiotis, L. Hadjileontiadis, and S. Panas. Pai-s/k: A robust automatic seismic p phase arrival identification scheme. IEEE Trans. Geosci. Remote Sens., 40:1395–1404, 2002.

[75] T. Sauer, J. Yorke, and M. Casdagli. Embedology. J. Stat. Phys., 65:579–616, 2000.

[76] A. Schclar, A. Averbuch, N. Rabin, V. Zheludev, and K. Hochman. A diffusion framework for detection of moving vehicles. Digit. Signal Process., 20:111–122., 2010.

[77] X. Shen and F.G. Meyer. Low-dimensional embedding of fMRI datasets. NeuroImage, 41(3):886 – 902, 2008.

[78] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22(8):888 –905, aug 2000.

[79] Y. Shimada, T. Kimura, and T. Ikeguchi. Analysis of chaotic dynamics using measures of the complex network theory. In ICANN '08: Proceedings of the 18th int. conf. on artificial neural networks, Part I, pages 61–70, Berlin, Heidelberg, 2008. Springer-Verlag.

[80] A. Singer, Y. Shkolnisky, and B. Nadler. Diffusion interpretation of nonlocal neighborhood filters for signal denoising. SIAM Journal of Imaging Sciences, 2(1):118–139, 2009.

[81] A. Szlam, M. Maggioni, and R. R. Coifman. Regularization on graphs with function-adapted diffusion processes. Journal of Machine Learning Research, 9:1711–1739, 2008.

[82] F. Takens. Detecting strange attractors in turbulence. In Dynamical systems and turbulence, Warwick 1980 (Coventry, 1979/1980), Lecture Notes in Math., 898, pages 366–381. Springer, 1981.

[83] K. M. Taylor. Sparse recovery and parameterization of manifold-valued data. M.S. thesis, May 2008.

[84] K. M. Taylor, M. J. Procopio, C. J. Young, and F. G. Meyer. Estimation of arrival times from seismic waves: a manifold-based approach. Geophysical Journal International, pages 435–452, 2011.

[85] V. Tejedor, O. Bénichou, and R. Voituriez. Global mean first-passage times of random walks on complex networks. Phys. Rev. E, 80(6):065104, Dec 2009.

[86] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. Science, 290(5500):2319–2323, December 2000.

[87] R. Tiwari, S. Srilakshmi, and K. Rao. Nature of earthquake dynamics in the central himalayan region: a nonlinear forecasting analysis. Journal of Geodynamics, 35:273–287, 2003.

[88] U. Trottenberg, C. W. Oosterlee, and A. Schuller. Multigrid. Academic Press, San Diego, CA, USA, 2001.

[89] R. Vautard and M. Ghil. Singular spectrum analysis in nonlinear dynamics, with applications to paleoclimatic time series. Physica D Nonlinear Phenomena, 35(3):395–424, 1989.

[90] A. Velasco, C. Young, and D. Anderson. Uncertainty in phase arrival time picks for regional seismic events: An experimental design. Technical report, US Department of Energy, 2001.

[91] S. Vempala. Geometric random walks: A survey. MSRI volume on Combinatorial and Computational Geometry, 2005.

[92] C. Wachinger and N. Navab. Manifold learning for multi-modal image registration. In Proceedings of the British Machine Vision Conference, pages 82.1–82.12. BMVA Press, 2010. doi:10.5244/C.24.82.

[93] M. B. Wakin, D. L. Donoho, H. Choi, and R. G. Baraniuk. The multiscale structure of non-differentiable image manifolds. In in Proc. Wavelets XI at SPIE Optics and Photonics, 2005.

[94] J. Wang. Adaptive training of neural networks for automatic seismic phase identification. Pure appl. Geophys., 159:1024–1041, 2002.

[95] M. Withers, R. Aster, C. Young, J. Beiriger, M. Harris, and S. Moore. comparison of select trigger algorithms for automated global seismic phase and event detection. Bull. seism. Soc. Am., 88:95–106, 1998.

[96] G. Yuan, M. Lozier, L. Pratt, C. Jones, and K. Helfrich. Estimating the predictability of an oceanic time series using linear and nonlinear methods. J. Geophys. Res, 2004.

[97] H. Zhang, C. Thurber, and C Rowe. Automatic p-wave arrival detection and picking with multiscale wavelet analysis for single-component recordings. Bull. Seism. Soc. Am., 93:1904, 2003.

[98] J. Zhang and M. Small. Complex network from pseudoperiodic time series: Topology versus dynamics. Phys. Rev. Lett., 96(23):238701, Jun 2006.

Appendix A

A.1 Proof of the lemma on the geometry of phase space

We will construct a matrix $\mathbf{F} : \mathbb{R}^d \to \mathbb{R}^{d-p}$ with $\mathbf{x}(t)$ in its nullspace. The matrix $\mathbf{F}$ will have entries that only depend on the roots of the ODE's characteristic equation. Furthermore, we will prove that $\mathbf{F}$ has full rank, and therefore the rank-nullity theorem asserts that the nullspace of $\mathbf{F}$ has dimension p.

Introduce the matrix

$$\mathbf{V}_{p\times d} = \begin{bmatrix} 1 & e^{r_1\Delta t} & e^{r_1 2\Delta t} & \dots & e^{r_1(d-1)\Delta t} \\ 1 & e^{r_2\Delta t} & e^{r_2 2\Delta t} & \dots & e^{r_2(d-1)\Delta t} \\ \vdots & & & & \\ 1 & e^{r_p\Delta t} & e^{r_p 2\Delta t} & \dots & e^{r_p(d-1)\Delta t} \end{bmatrix}.$$

Now, observe that if $d = p$, then

$$\det \mathbf{V}_{p\times d} = \prod_{1\le i<j\le p} (e^{r_j\Delta t} - e^{r_i\Delta t}).$$

Therefore, $\mathbf{V}_{p\times p}\mathbf{n} = 0$ has only the trivial solution. Moreover

$$\dim(\ker(\mathbf{V}_{p\times(p+k)})) = k.$$

Now, define $\mathbf{n}^* = (n_1^*, \dots, n_{p+1}^*)^T \in \mathbb{R}^{p+1}$ as a unit vector spanning the one-dimensional nullspace of $\mathbf{V}_{p\times(p+1)}$ and use its entries to populate rows of the $(d-p)$-by-d matrix

$$\mathbf{F}_{(d-p)\times d} = \begin{bmatrix} n_1^* & \cdots & n_{p+1}^* & 0 & 0 \\ 0 & n_1^* & \cdots & n_{p+1}^* & 0 \\ & & \ddots & & \\ 0 & 0 & n_1^* & \cdots & n_{p+1}^* \end{bmatrix}. \tag{A.1}$$

Observe that we can write the i^{th} entry of the matrix-vector product $\mathbf{F}_{(d-p)\times d}\mathbf{x}(t)$ as

$$\begin{aligned} \left(\mathbf{F}_{(d-p)\times d}\mathbf{x}(t)\right)_i &= n_1^*\, x(t+(i-1)\Delta t) + n_2^*\, x(t+i\Delta t) + \cdots + n_{p+1}^*\, x(t+(i+p-1)\Delta t) \\ &= n_1^* \left(\sum_{j=1}^{p} \alpha_j e^{r_j(t+(i-1)\Delta t)}\right) + \cdots + n_{p+1}^* \left(\sum_{j=1}^{p} \alpha_j e^{r_j(t+(i+p-1)\Delta t)}\right) \\ &= \sum_{j=1}^{p} \left(n_1^* + n_2^*\, e^{r_j\Delta t} + \cdots + n_{p+1}^*\, e^{r_j p\Delta t}\right)\, \alpha_j\, e^{r_j(i-1)\Delta t}\, e^{r_j t}. \end{aligned}$$

The inner sum vanishes by construction of $\mathbf{n}^*$. It follows that $\mathbf{F}\mathbf{x}(t) = 0$ for all t. Finally, it is clear that $\mathbf{F}_{(d-p)\times d}$ has linearly independent rows, hence, $\mathbf{F}_{(d-p)\times d}$ is full rank. Finally, the rank-nullity theorem asserts that the nullity of $\mathbf{F}_{(d-p)\times d}$ is p.

A.1.1 Proof of Corollary 1 — constituent frequencies

Using complex exponentials, it is clear that $x(t)$ is solution to a differential equation of the form (3.5) of order $2K$ with characteristic equation $\prod_{k=1}^{K}(r - i\,\omega_k)(r + i\,\omega_k)$, where $i = \sqrt{-1}$. Application of Proposition 1 completes the proof.

A.1.2 Proof of Corollary 2 — local approximations

Let $\mathbf{Q}_2$ be the orthogonal projection onto the p-dimensional subspace containing solutions to (3.5). It follows that the minimum Euclidean distance between the trajectory and the subspace is given by

$$\|(\mathbf{I} - \mathbf{Q}_2)\mathbf{x}(t)\| = \left\|(\mathbf{I} - \mathbf{Q}_2)\left(\sum_{i=1}^{p} b_i \mathbf{y}_i(t) + \mathbf{e}(t)\right)\right\|,$$

where $\mathbf{y}_i(t) = (y_i(t), \ldots, y_i(t + (d-1)\Delta t)^T$ is the delay-coordinate embedding of $y_i(t)$, $\mathbf{e}(t) = (e(t), \ldots, e(t + (d-1)\Delta t)^T$ is the delay-coordinate embedding of $e(t)$, and b_i are expansion coeffi-

cients. Since $\mathbf{Q}_2\mathbf{y}_i(t) = \mathbf{y}_i(t)$, and since the operator norm of the orthogonal projection $(\mathbf{I} - \mathbf{Q}_2)$ is at most one, we have that

$$\begin{aligned}\|(\mathbf{I} - \mathbf{Q}_2)\mathbf{x}(t)\| &= \|(\mathbf{I} - \mathbf{Q}_2)\mathbf{e}(t)\| \\ &\leq \|\mathbf{e}(t)\|.\end{aligned}$$

A.2 A possible direction on the conjecture on the dimensionality of image patch-sets

We begin by thinking of the patch $\mathbf{p}(x, y)$ (defined in (3.8)) as a vector in $\mathbb{R}^{d^2}$. As in the proof of Appendix A.1, our goal is to construct a matrix $\mathbf{F}$: with $\mathbf{x}(t)$ in its null-space, and then show that the rank of this matrix is $d^2 - pq$. The difficulty lies in proving the matrices rank, but we will describe its construction.

Let the characteristic equation of the ODEs (3.10) and (3.11) be given by

$$\Xi_X(r) = \prod_{k=1}^{p}(r - \lambda_k), \qquad \text{and} \qquad \Xi_Y(r) = \prod_{j=1}^{q}(r - \mu_j),$$

respectively. It follows that the solutions can be written as

$$X(x) = \sum_{k=1}^{p} \alpha_k e^{\lambda_k x}, \qquad \text{and} \qquad Y(y) = \sum_{l=1}^{q} \beta_l e^{\mu_l y},$$

where the coefficients α_k and β_j are chosen to satisfy initial or boundary conditions.

We will write a vector $\mathbf{v} \in \mathbb{R}^{d^2}$ as a matrix with entries v_{ij}:

$$\mathbf{v} = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1d} \\ v_{21} & v_{22} & & \\ & & \ddots & \\ v_{d1} & v_{d2} & \dots & v_{dd} \end{bmatrix}.$$

It follows that if we also regard the patch $\mathbf{p}(x, y)$ as a vector, then the inner product between $\mathbf{p}(x, y)$ and $\mathbf{v}$ can be written as

$$\langle \mathbf{p}(x, y), \mathbf{v} \rangle = \sum_{j=1}^{d} Y(y + (j-1)\Delta y) \sum_{i=1}^{d} v_{ij} X(x + (i-1)\Delta x).$$

Observe that

$$\sum_{i=1}^{d} v_{ij} X(x + (i-1)\Delta x) = \sum_{k=1}^{p} \left(\sum_{i=1}^{d} v_{ij} e^{\lambda_k (i-1)\Delta x} \right) e^{\lambda_k x}.$$

So

$$\langle \mathbf{p}(x,y), \mathbf{v} \rangle = \sum_{j=1}^{d} Y(y + (j-1)\Delta y) \sum_{k=1}^{p} \left(\sum_{i=1}^{d} v_{ij} e^{\lambda_k (i-1)\Delta x} \right) e^{\lambda_k x}. \tag{A.2}$$

An equivalent argument where the roles of $X(x)$ and $Y(y)$ are switched yields

$$\langle \mathbf{p}(x,y), \mathbf{v} \rangle = \sum_{i=1}^{d} X(x + (i-1)\Delta x) \sum_{l=1}^{q} \left(\sum_{j=1}^{d} v_{ij} e^{\mu_l (j-1)\Delta y} \right) e^{\mu_l y}. \tag{A.3}$$

Now, consider the matrices

$$\mathbf{V}_{p\times d} = \begin{bmatrix} 1 & e^{\lambda_1 \Delta t} & e^{\lambda_1 2\Delta t} & \dots & e^{\lambda_1 (d-1)\Delta t} \\ 1 & e^{\lambda_2 \Delta t} & e^{\lambda_2 2\Delta t} & \dots & e^{\lambda_2 (d-1)\Delta t} \\ & & & \ddots & \\ 1 & e^{\lambda_p \Delta t} & e^{\lambda_p 2\Delta t} & \dots & e^{\lambda_p (d-1)\Delta t} \end{bmatrix},$$

and

$$\mathbf{W}_{q\times d} = \begin{bmatrix} 1 & e^{\mu_1 \Delta t} & e^{\mu_1 2\Delta t} & \dots & e^{\mu_1 (d-1)\Delta t} \\ 1 & e^{\mu_2 \Delta t} & e^{\mu_2 2\Delta t} & \dots & e^{\mu_2 (d-1)\Delta t} \\ & & & \ddots & \\ 1 & e^{\mu_q \Delta t} & e^{\mu_q 2\Delta t} & \dots & e^{\mu_q (d-1)\Delta t} \end{bmatrix}.$$

Note that if $d = p$, then

$$\det \mathbf{V}_{p\times p} = \prod_{1\le i<j\le p} (e^{\lambda_j \Delta x} - e^{\lambda_i \Delta x}),$$

and if $d = q$, then

$$\det \mathbf{W}_{q\times q} = \prod_{1\le i<j\le q} (e^{\mu_j \Delta y} - e^{\mu_i \Delta y}).$$

Because the roots are simple by assumption, the matrices $\mathbf{V}_{p\times p}$ and $\mathbf{W}_{q\times q}$ are full rank. Moreover, the nullspaces of the matrices $\mathbf{V}_{p\times(p+1)}$ and $\mathbf{W}_{q\times(q+1)}$ are one-dimensional.

We define $\mathbf{v}^* = (v_1^*, \dots, v_{p+1}^*)^T \in \mathbb{R}^{p+1}$ as a unit vector spanning the one-dimensional nullspace of $\mathbf{V}_{p\times(p+1)}$, and $\mathbf{w}^* = (w_1^*, \dots, w_{q+1}^*)^T \in \mathbb{R}^{q+1}$ as a unit vector spanning the one-dimensional

null-space of $\mathbf{W}_{q\times(q+1)}$. It follows that we can use $\mathbf{v}^*$ and $\mathbf{w}^*$ to make the inner sums of the inner products (A.2) and (A.3) vanish. Using the same construction of F as in Appendix A.1, we would obtain a matrix F of size $d(d-p)+d(d-q)$. The factors $(d-p)$ and $(d-q)$ are analogous to the size of the matrix in one-dimension; The factor of d accounts for the number of one-dimensional slices of the patch that we can take in the horizontal direction.

Although an analytic proof seems intractable, numerical and symbolic experiments indicate that the resulting matrix always has rank d^2-pq. This would suggest the nullspace has dimension pq, which would complete the proof.

A.3 Relating mean-subtraction to local-mean-oscillation

Assume that $x(t)$ is locally integrable and define the local mean of $x(t)$ over the interval $J_0=\{\tau\in\mathbb{R}:|t-\tau|<\epsilon_0\}$ as

$$\overline{x}(t)=\frac{1}{2\epsilon_0}\int_{t-\epsilon_0}^{t+\epsilon_0}x(\theta)d\theta.$$

The local-mean-oscillation of $x(t)$ on J_0 is defined as

$$L_{x,J_0}(t)=\frac{1}{2\epsilon_0}\int_{t-\epsilon_0}^{t+\epsilon_0}|x(\tau)-\overline{x}(t)|\,d\tau. \tag{A.4}$$

We will show that at each fixed time t, the minimum distance between the trajectory $\mathbf{x}(t)$ and the subspace spanned by the vector $(1,1,\ldots,1)^T\in\mathbb{R}^d$ is related to the local-mean-oscillation of the signal on the set $J_1=\{\tau:t\le\tau\le t+(d-1)\Delta t\}$. Let $x_k(t)=x(t+(k-1)\Delta t)$ for $k=1,2,\ldots,d$. Approximating (A.4) using the composite midpoint rule for integration over the set J_1 with nodes at times $t_k=t+(k-1)\Delta t$ for $k=1,2,\ldots,d$ leads to the estimate

$$\widetilde{L}_{x,J_1}(t)=\frac{1}{d}\sum_{k=1}^{d}\left|x_k(t)-\tilde{\overline{x}}(t)\right|, \tag{A.5}$$

where $\tilde{\overline{x}}(t)=\frac{1}{d}\sum_{j=1}^{d}x_j(t)$.

Now, define $\mathbf{Q}_0$ as the orthogonal projection of $\mathbb{R}^d$ onto the subspace spanned by the constant vector $(1,1,\ldots,1)^T\in\mathbb{R}^d$ so that $\mathbf{Q}_0\mathbf{x}(t)=\tilde{\overline{x}}(t)(1,1,\ldots,1)^T=\left(\frac{1}{d}\sum_{j=1}^{d}x_j(t)\right)(1,1,\ldots,1)^T$.

Finally, observe that we can write

$$\widetilde{L}_{x,J_1}(t) = \tfrac{1}{d}\langle \mathbf{x}(t) - \mathbf{Q}_0\mathbf{x}(t), \mathbf{v}\rangle,$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean inner product, and $\mathbf{v} \in \mathbb{R}^d$ is a vector whose k^{th} entry is given by

$$v_k = \text{sign}\left(x_k(t) - \tilde{\overline{x}}(t)\right) \in \{\pm 1\}.$$

It follows that

$$\begin{aligned}
\widetilde{L}_{x,J_1}(t) &= \frac{1}{d}\langle \mathbf{x}(t) - \mathbf{Q}_0\mathbf{x}(t), \mathbf{v}\rangle \\
&\leq \frac{1}{d}\|\mathbf{x}(t) - \mathbf{Q}_0\mathbf{x}(t)\|\|\mathbf{v}\| \\
&= \frac{1}{\sqrt{d}}\|\mathbf{x}(t) - \mathbf{Q}_0\mathbf{x}(t)\|,
\end{aligned}$$

where the second inequality follows after application of the Cauchy-Schwartz inequality. Hence, the Euclidean distance $\|\mathbf{x}(t) - \mathbf{Q}_0\mathbf{x}(t)\|$ can be thought of as an approximation that always over-estimates $\sqrt{d}$ factors of an estimate of the local-mean-oscillation of $x(t)$ on the set J_1.

A.4 A note on the frequency content in a patch after normalizing

In this section, we consider normalizing a patch by subtracting away its mean, or by fixing its ℓ^2 norm. We will demonstrate in what sense these normalizations will preserve frequency information in the patch-set.

Let $x_j(t) = x(t + j\Delta)$ and let the mean of the coordinates defining the trajectory at time t be $\overline{x}(t) = \frac{1}{d}\sum_{j=0}^{d-1} x_j(t)$. At a fixed time t_n, $\overline{x}(t_n) = \overline{x}_n$ is average value in a patch, and we can interpret this as a convolution of discrete sequences $(\mathbf{x}_n * h_n)$, where h_n is the impulse response defined by

$$h_n = \begin{cases} \frac{1}{d} & \text{if } n \in \{0, 1, 2, \ldots, (d-1)\}, \\ 0 & \text{else.} \end{cases}$$

The associated frequency response is given by

$$\hat{h}_\omega = \frac{1 - e^{-id\omega}}{d(1 - e^{-i\omega})}.$$

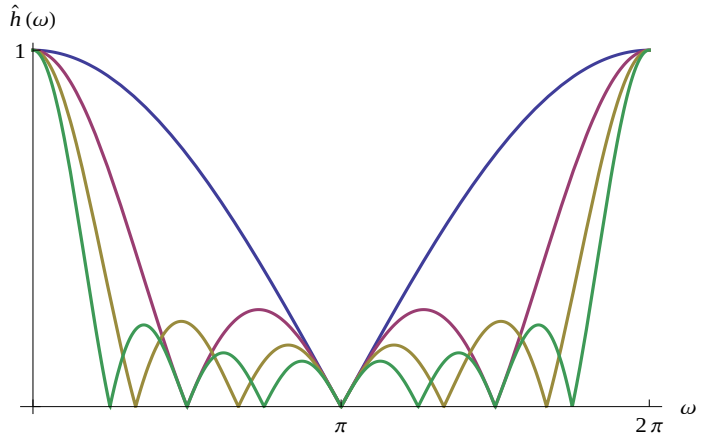

Figure A.1: The magnitude of the frequency response of an averaging filter for various patch sizes d. The blue, red, yellow, and green curves correspond to the parameter $d = 2, 4, 6, 8$, respectively.

The magnitude of $\hat{h}_\omega$ is given in Figure A.1 for $d \in \{2, 4, 6, 8\}$. As d increases, the convolution passes lower frequencies and attenuates higher frequencies. Note that frequencies $\omega \geq \pi$ are aliased to $\omega' = \omega - \pi$. Consequently, the sequence $\overline{x}_n$ preserves the low-frequency information in the patch $\mathbf{x}_n$. If we introduce $\overline{\mathbf{x}} = (\overline{x}_n, \ldots, \overline{x}_n) \in \mathbb{R}^d$, then the difference

$$\mathbf{x}_n - \overline{\mathbf{x}}_n$$

resembles $\mathbf{x}_n$ with low-frequency global trends removed and high-frequency information preserved. This is helpful, since we suspect local irregularities in $x(t)$ to be characterized by rapid frequency variations (equivalently, large magnitude derivatives).

Now, introduce the function

$$(s(t))^2 = \frac{1}{d}\sum_{j=0}^{d-1}(x_j(t) - \overline{x}(t))^2,$$

and let $\tilde{x}_j(t) = (x_j(t) - \overline{x}(t))$. The Fourier transform of the ratio $\tilde{x}_j(t)/s(t)$ is given by

$$\widehat{\left(\frac{\tilde{x}_j}{s}\right)}(\omega) = \widehat{(\tilde{x}_j\,\tilde{s})}(\omega) = \left(\widehat{\tilde{x}}_j * \widehat{\tilde{s}}\right)(\omega),$$

where $\tilde{s}(t) = (s(t))^{-1}$. It follows that the frequency content in the ratio is determined by convolving the frequency content of $\tilde{x}_j(t)$ with the frequency content of $\tilde{s}(t)$, which also resembles filtering.

Therefore, if $s(t)$ is sufficiently differentiable, then the Fourier Transform of $\tilde{s}(t)$ will be localized around the origin, and the frequency content in the ratio $\tilde{x}_j(t)/s(t)$ can be approximated as

$$\left(\widehat{\tilde{x}_j} * \widehat{\tilde{s}}\right)(\omega) \approx \left(\widehat{\tilde{x}_j} * \delta\right)(\omega) = \widehat{\tilde{x}_j}(\omega).$$

Hence, the frequency content in each component function $\tilde{x}_j(t)$ will be preserved when we normalize by dividing by the standard deviation of values in a local temporal neighborhood.

A.5 The connectedness of the fast graph model

It is necessary that the fast graph $\mathcal{F}(N,p)$ be connected for any of the theory behind the spectral representation of the commute-time parametrization to hold. To ensure that the probability of $\mathcal{F}(N,p)$ being disconnected will vanish as N gets large, we must choose $Np > \log N$ [30]. Since p is defined as a function of L in (4.12), any requirement on p ultimately constrains L. First, because the maximum degree of a vertex in $\mathcal{S}(N,L)$ is $2L+1$, according to (4.10), we require

$$2L + 1 \leq N.$$

Manipulation of this inequality leads to

$$\frac{L+1}{N} \leq \frac{1}{2} + \frac{1}{2N}.$$

We assume that $N \geq 2$, so that

$$\frac{L+1}{N} \leq \frac{3}{4}.$$

It follows that

$$\left(2 - \frac{L+1}{N}\right) \geq \frac{5}{4} > 1.$$

Therefore, rewriting (4.12) and using the last inequality we have

$$\begin{aligned} p &= \frac{L}{N-1}\left(2 - \frac{L+1}{N}\right) \\ &> \frac{L}{N}\left(2 - \frac{L+1}{N}\right) \\ &> \frac{L}{N}. \end{aligned}$$

Therefore, choosing $L = c \log N$ for some $c > 1$ ensures that $Np > \log N$, and consequently, the probability of $\mathcal{F}(N, p)$ being disconnected approaches zero as N approaches infinity.

A.6 Bounding the commute-times in the graph models

A.6.1 Proof of the lower bound on the average commute-time in the slow graph

Before presenting the proof, we discuss two alternative approaches for determining a lower bound that we do not pursue because of their apparent difficulty. These approaches are based on the special structure of the weight matrix associated with the slow graph model. In particular, because the weight matrix is Toeplitz, one may try to analytically obtain the eigenvectors and eigenvalues used in the spectral representation of the commute-time (4.3), or easily solve the linear system that results when performing one-step analyses. We now point out why we avoid such approaches.

First, consider performing one-step analyses [55] in order to compute the commute-time as a sum of hitting times $\eta(\mathbf{x}_n, \mathbf{x}_m)$ and $\eta(\mathbf{x}_m, \mathbf{x}_n)$. It is well known that the hitting time satisfies the equation

$$\eta(\mathbf{x}_n, \mathbf{x}_m) = 1 + \sum_{k=1}^{N} (\mathbf{P})_{nk} \eta(\mathbf{x}_k, \mathbf{x}_m),$$

where $(\mathbf{P})_{nk}$ is (n, k) entry of the probability transition matrix $\mathbf{P}$ introduced in section 4.2. For fixed $m \in \{1, 2, \ldots, N\}$, one would solve the above system for the unknowns $\eta(\mathbf{x}_n, \mathbf{x}_m)$, where $n \in \{1, 2, \ldots, m-1, m+1, \ldots, N\}$, subject to the condition that $\eta(\mathbf{x}_m, \mathbf{x}_m) = 0$. Then, one would have to solve another linear system, where the roles of $\mathbf{x}_n$ and $\mathbf{x}_m$ are reversed. In total, one would have to solve $\lceil N/2 \rceil$ distinct linear systems, due to the symmetry of the slow graph. We do not use one-step analyses because of the large number of matrices that must be considered. Notice that, for nodes $\mathbf{x}_n$ that are not connected directly to $\mathbf{x}_m$, the coefficients of the linear equations defining $\eta(\mathbf{x}_n, \mathbf{x}_m)$ will have counterparts in the matrix

$$\mathbf{D}^{-1/2}(\mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2})\mathbf{D}^{1/2}.$$

This observation is additional motivation for expressing the commute-time using the spectral decomposition of $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$, rather than solve every linear system associated with one-step analyses.

Now, consider computing the eigenvectors and eigenvalues used in the spectral representation of the commute-time, defined in (4.3). Because the matrix $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ is Toeplitz, and not circulant, we cannot, in general, diagonalize it with the Fourier transform when N is finite. As an alternative, one would have to approximate $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ with sequence of circulant matrices. Then, one could obtain some handle on the asymptotic properties of the eigenvectors and eigenvalues. We do not take this approach because it again requires considering so many different matrices. Furthermore, we would like to obtain estimates for finite N. As another alternative, one might try to approximate the slow-graph as a $(2L+1)$-regular ring-lattice, which has a circulant weight matrix. Then, one could argue that the commute-times on the ring-lattice are less than the commute-times on the slow-graph, thereby obtaining a lower bound. Although such an approach would provide analytic expressions for λ_k and ϕ_k used in the spectral representation of the commute-time expansion, we do not take this approach because the terms become so complicated that it appears the best lower bound we can obtain is the trivial lower bound given by

$$\frac{1}{2}\left(\frac{1}{\pi_n}+\frac{1}{\pi_m}\right) \leq \kappa(\mathbf{x}_n, \mathbf{x}_m),$$

where we have used orthogonality of the ϕ_k, and the fact that $(1-\lambda_k)^{-1} \geq 1/2$. Since $\pi_n = N^{-1}$ on the regular ring-lattice, the left hand side grows like N, which is not fast enough to prove Corollary 1.

In light of the above difficulties, we consider a third approach.

In order to compute a lower bound on the average commute time, we consider a fixed pair of vertices in the slow graph, $\mathbf{x}_{n_0}$ and $\mathbf{x}_{m_0}$, and compute a lower bound on the commute time $\kappa(\mathbf{x}_{n_0}, \mathbf{x}_{m_0})$. We can then compute the average of this lower bound over all the pairs of vertices.

To obtain the lower bound on $\kappa(\mathbf{x}_{n_0}, \mathbf{x}_{m_0})$ we use a standard tool to obtain lower bounds on commute time: the Nash-Williams inequality [57]. The Nash-Williams inequality is usually

formulated in terms of electrical networks. We prefer to present an equivalent formulation that is directly adapted to our problem. We first introduce the concept of an *edge-cutset*.

Definition 11. *Let V_1 and V_2 be two disjoint sets of vertices. A set of edges E is an edge-cutset separating V_1 and V_2 if every path that connects a vertex in V_1 with a vertex in V_2 includes an edge in E.*

Given a weighted graph, which may contain loops, we define a random walk with the probability transition matrix $\mathbf{P}_{n,m} = \mathbf{W}_{n,m}/\mathbf{D}_{n,n}$. Let $\mathbf{x}_{m_0}$ and $\mathbf{x}_{n_0}$ be two vertices. The commute time between vertices $\mathbf{x}_{m_0}$ and $\mathbf{x}_{n_0}$, $\kappa(\mathbf{x}_{m_0}, \mathbf{x}_{n_0})$ satisfies the following lower bound.

Lemma 3 (Nash-Williams)**.** *If $\mathbf{x}_{m_0}$ and $\mathbf{x}_{n_0}$ are distinct vertices in a graph that are separated by disjoint edge-cutsets $E_k, k = 1, \ldots,$ then*

$$V \sum_k \left[\sum_{\{\mathbf{x}_n, \mathbf{x}_m\} \in E_k} w_{n,m} \right]^{-1} \leq \kappa(\mathbf{x}_{m_0}, \mathbf{x}_{n_0}) \qquad \text{where } \{\mathbf{x}_m, \mathbf{x}_n\} \text{ is an edge in the cutset } E_k, \tag{A.6}$$

and where the volume of the graph is defined by $V = \sum_{i=1}^{N} \sum_{j=1}^{N} w_{i,j}$.

We now exhibit a sequence of edge-cutsets in the slow graph. We refer to Figure A.2 for the construction of the cutsets. We define the first cutset E_1. If $m_0 < L$, then E_1 needs a little more attention and is defined as the set of L edges $\{\mathbf{x}_i, \mathbf{x}_j\}$, where i and j are defined by

$$\begin{cases} i = 1, \ldots, m_0, \\ j = m_0 + 1, \ldots, L + i. \end{cases} \tag{A.7}$$

The edge-cutset E_1 is shown in the Figure A.2 for $m_0 = 1$ (left) and $m_0 = 2$ (center), for $L = 3$. The removal of this set of edges prevents $\mathbf{x}_{m_0}$ from being connected to $\mathbf{x}_{n_0}$. Indeed, the self loop on the diagonal (green entry) does not allow the random walk to move toward $\mathbf{x}_{n_0}$. This can be also be visualized in Figure A.3, where E_1 is the leftmost set of edges that connect $\mathbf{x}_{m_0}$ to that part of the graph that is connected to $\mathbf{x}_{n_0}$. The sum of edge weights in E_1 is at most $L(L+1)w_{\mathcal{S}}/2$.

If $m_0 \geq L$, then E_1, is defined as the other generic edge-cutsets.

We now define the generic edge-cutsets E_k as the set of $L(L+1)/2$ edges $\{\mathbf{x}_i, \mathbf{x}_j\}$ such that

$$\begin{cases} i = m_0 + 1 + (k-2)L, \ldots, m_0 + (k-1)L, \\ \\ j = m_0 + 1 + (k-1)L, \ldots, L + i. \end{cases} \tag{A.8}$$

As seen in Figure A.2-right for $k = 3$, setting the entries of E_3 to zero disconnects the upper and lower part of the submatrix $\mathbf{W}(m_0 : n_0, m_0 : n_0)$, thereby isolating $\mathbf{x}_{m_0}$ and $\mathbf{x}_{n_0}$. Alternatively, we also see in Figure A.3 that any path from $\mathbf{x}_{m_0}$ to $\mathbf{x}_{n_0}$ needs to go through E_3. Each edge-cutset $E_k, k \geq 2$ is a triangle with a height of size L. Therefore, after creating E_1, we can fit $\left\lfloor \frac{n_0-(m_0+1)+1}{L} \right\rfloor$ such cutsets between $\mathbf{x}_{m_0+1}$ and $\mathbf{x}_{n_0}$. The sum of the weights along the edges of each cutset $E_k, k = 2, \ldots$ is given by $L(L+1)w_s/2$. In addition, we have the sum of edge weights in the first cutset E_1 is at most $L(L+1)w_s/2$. Putting everything together, the computation of the lower bound using the Nash-Williams Lemma yields

$$\begin{aligned} & V \sum_k \left[\sum_{\{n,m\} \in E_k} w_{n,m} \right]^{-1} \\ & \quad \geq \left[N(2L+1) - L(L+1) \right] w_s \left(\left\lfloor \frac{(n_0 - m_0)}{L} \right\rfloor \frac{2}{L(L+1)w_s} + \frac{2}{L(L+1)w_s} \right) \\ & \quad \geq \frac{\left[N(2L+1) - L(L+1) \right]}{L(L+1)} \left(2 \left(\frac{n_0 - m_0}{L} - 1 \right) + 2 \right) \\ & \quad \geq \frac{\left[N(2L+1) - L(L+1) \right]}{L(L+1)} \left(2 \frac{n_0 - m_0}{L} \right) \end{aligned}$$

We can summarize this result in the following lemma.

Lemma 4. *The commute time between vertices $\mathbf{x}_{n_0}$ and $\mathbf{x}_{m_0}$ inside $\mathcal{S}(N, L)$ satisfies*

$$\kappa(\mathbf{x}_{m_0}, \mathbf{x}_{n_0}) \geq \frac{2\left[N(2L+1) - L(L+1) \right]}{L(L+1)} \left(\frac{n_0 - m_0}{L} \right). \tag{A.9}$$

Finally, we bound the average commute time in the slow graph. Observe that the slow graph model $\mathcal{S}(N, L)$ has $N - j$ pairs of vertices such that $|m - n| = j$, for $j = 1, \ldots, N-1$. Therefore, using the lower bound given in Lemma 4 it follows that

$$\sum_{1 \leq m < n \leq N} \kappa(\mathbf{x}_m, \mathbf{x}_n) \geq \frac{2\left[N(2L+1) - L(L+1) \right]}{L^2(L+1)} \sum_{j=1}^{N-1} (N-j)j$$

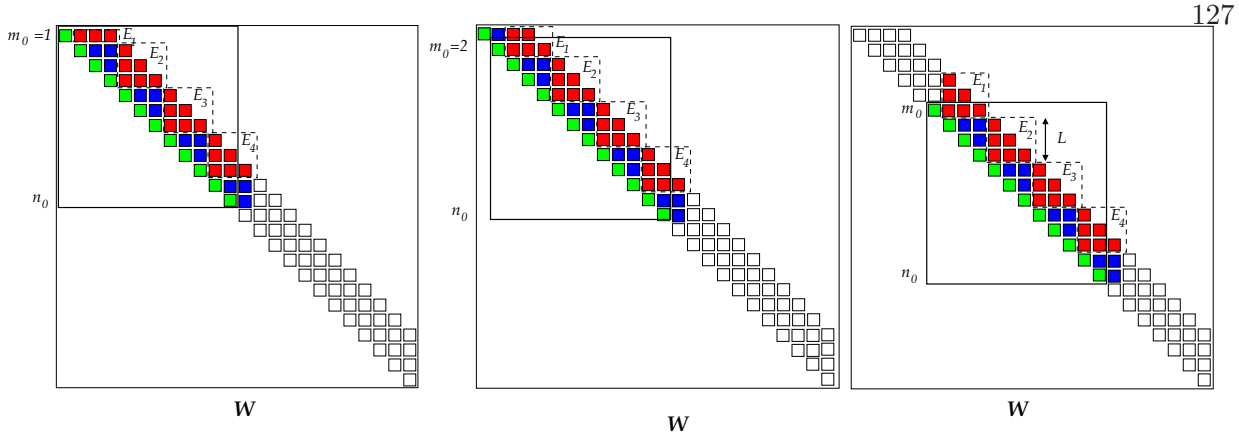

Figure A.2: Each small square represents a nonzero entry in the upper triangular portion of the weight matrix $\mathbf{W}$ of $\mathcal{S}(N, L)$. The submatrix $\mathbf{W}(m_0 : n_0, m_0 : n_0)$ is also shown The green entries on the diagonal are the self-loops. The edge-cutsets E_k are shown in red for $m_0 = 1$ (left), $m_0 = 2$ (center), and for $m_0 \geq L$ (right).

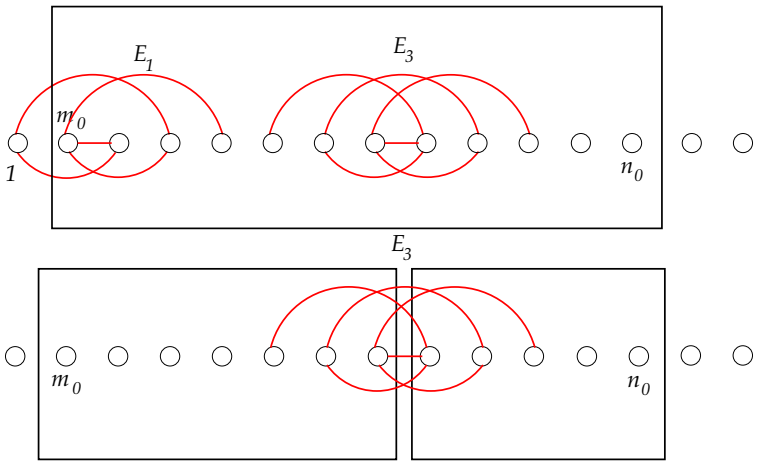

Figure A.3: Top: edge-cutsets E_1 and E_3. Bottom: any path from m_0 to n_0 needs to use an edge of the edge-cutset E_3.

But

$$\sum_{j=1}^{N-1}(N-j)j = \left(N\sum_{1}^{N-1} j - \sum_{1}^{N-1} j^2\right) = \left(\frac{N^2(N-1)}{2} - \frac{N(N-1)}{2}\frac{2N-1}{3}\right)$$
$$= \frac{N(N-1)}{2}\frac{N+1}{3}$$

Dividing both sides by $N(N-1)/2$ and simplifying yields (4.15).

A.6.2 Proof of upper bound on the average commute time in the fast graph

Our approach relies on the relationship between electrical networks and random walks on graphs [29]. We begin by introducing the property of interest — the *effective resistance* — and its relationship to the commute time.

The electrical network perspective For each pair of vertices $\mathbf{x}_n$ and $\mathbf{x}_m$ with a non zero weight $w_{n,m}$, we assign the resistance

$$r_{n,m} = \frac{1}{w_{n,m}} \tag{A.10}$$

to the edge $\{\mathbf{x}_n, \mathbf{x}_m\}$. We note that if $w_{n,m} = 0$, then there is no connection between $\mathbf{x}_n$ and $\mathbf{x}_m$, and no resistance to consider. Now, consider applying a potential difference, or voltage, across the vertices $\mathbf{x}_{m_0}$ and $\mathbf{x}_{n_0}$. As a result, some current flows across the resistors (edges) in the electrical network (graph). We may replace the set of resistors across which some current flows by an equivalent, *effective resistance*, R_{m_0,n_0} that is connected between $\mathbf{x}_{m_0}$ and $\mathbf{x}_{n_0}$. The effective resistance R_{m_0,n_0} is defined by the voltage necessary to maintain a one-unit current between $\mathbf{x}_{m_0}$ and $\mathbf{x}_{n_0}$. The main result in [17], is that the commute time between vertices $\mathbf{x}_{m_0}$ and $\mathbf{x}_{n_0}$ can be expressed as

$$\kappa(\mathbf{x}_{m_0}, \mathbf{x}_{n_0}) = V R_{m_0,n_0}. \tag{A.11}$$

Taking expectations of both sides of Equation (A.11) with respect to the process of generating edges and choosing terminals in a fast graph, we obtain

$$\kappa_{\mathcal{F}} = \mathrm{E}(V)\,\mathrm{E}(R) + \mathrm{Cov}(V, R). \tag{A.12}$$

Notice that every edge in the fast graph has weight $w_{\mathcal{F}}$. Therefore, V can be expressed as

$$V = \sum_{n=1}^{N} w_{nn} + 2 \sum_{1 \le m < l \le N} w_{ml} = w_{\mathcal{F}} N + 2 w_{\mathcal{F}} \tilde{N}, \tag{A.13}$$

where $\tilde{N}$ is a binomial random variable representing the number of edges connecting distinct vertices in the fast graph. We now rewrite (A.12), using (A.13) and the assumption that $\mathrm{Cov}(V, R) \le 0$, to obtain

$$\kappa_{\mathcal{F}} \leq w_{\mathcal{F}} \left[N + 2\mathrm{E}(\tilde{N}) \right] \mathrm{E}(R).$$

Recall that $\tilde{N}$ is distributed as a binomial random variable with parameters $(N(N-1)/2, p)$. Also, the effective resistance between two nodes of a network is at most the geodesic distance between them, δ, scaled by $1/w_{\mathcal{F}}$ [17]. It follows that

$$\kappa_{\mathcal{F}} \leq [N(N-1)p + N] \, \mathrm{E}(\delta).$$

The authors [38] give a closed form expression for $\mathrm{E}(\delta)$ on Erdös-Renyi graphs, which we can utilize since the fast graph's self-connections do not change the geodesic distance. This yields

$$\kappa_{\mathcal{F}} \leq [N(N-1)p + N] \left[\frac{\log N - \gamma_e}{\log((N-1)p + 1)} + \frac{1}{2} \right],$$

where $\gamma_e \approx 0.5772$ is Euler's constant. Simplification using (4.12) gives the desired result.

Remark Although $\mathrm{Cov}(V, R) \leq 0$ is an assumption, we conjecture that it is always satisfied due to the fact that increasing the number of resistors M in an electrical network with a fixed number of nodes is effectively like adding resistors-in-parallel, and, according to Rayleigh's Monotonicity Law, adding edges (increasing M) can only decrease the effective resistance [29].

A.7 Generating a random trigonometric polynomial with a specified autocorrelation

Let $z(t)$ represent a random trigonometric polynomial on $[0, 1)$ with an autocorrelation function given by

$$C(\tau) = 2(\cos(\pi\tau))^{2\beta} - 1 \qquad \text{for } \tau \in \left[-\frac{1}{2}, \frac{1}{2} \right), \tag{A.14}$$

for some nonnegative integer β. It follows that we can do a Fourier expansion of $C(\tau)$ to obtain

$$C(\tau) = \sum_{j \in \mathbb{Z}} \hat{C}_j e^{2\pi i j t}, \tag{A.15}$$

where $i = \sqrt{-1}$ and

$$\begin{aligned}
\hat{C}_j &= \int_0^1 C(\tau) e^{-2\pi i j \tau} d\tau \\
&= \int_0^1 \left(2^{(1-2\beta)} \left(\sum_{k=0}^{2\beta} \binom{2\beta}{k} e^{2\pi i(\beta-k)\tau} \right) - 1 \right) e^{-2\pi i j \tau} d\tau \\
&= 2^{(1-2\beta)} \sum_{k=0}^{2\beta} \binom{2\beta}{k} \int_0^1 e^{2\pi i(\beta-k-j)\tau} \, d\tau - \int_0^1 e^{-2\pi i j \tau} \, d\tau \\
&= \begin{cases} 2^{(1-2\beta)} \binom{2\beta}{\beta} - 1 & \text{if } j = 0, \\ 2^{(1-2\beta)} \binom{2\beta}{\beta - j} & \text{if } |j| \leq \beta, \\ 0 & \text{if } j > \beta, \end{cases}
\end{aligned}$$

where the second equality follows after expressing cosine with complex exponentials, and applying the binomial theorem.

It is clear that 2β is the frequency of the fastest sinusoid making up the random signal $z(t)$, and that most of the energy is on average at frequency β. Let A_j and B_j be independent and identically distributed Normal random variables with zero mean and unit variance. Define

$$\hat{z}_j = \sqrt{\frac{\hat{C}_j}{2}} \left(A_j + i B_j \right).$$

Finally, the signal $z(t)$ is defined as

$$z(t) = \sum_{j \in \mathbb{Z}} \hat{z}_j e^{2\pi i j t}.$$

To check that the signal $z(t)$ defined above has the correct autocorrelation, observe that linearity of the expectation, independence and zero mean of the random variables, and the fact that $\hat{C}_j = \hat{C}_{-j}$

together imply that

$$\begin{aligned}
\mathrm{E}(z(t)\overline{z(t+\tau)}) &= \sum_{|j|\leq 2\beta}\sum_{|k|\leq 2\beta} \mathrm{E}\left(\hat{z}_j\overline{\hat{z}_k}\right) e^{-2\pi ik\tau}e^{2\pi i(j-k)t} \\
&= \sum_{|j|\leq 2\beta}\sum_{|k|\leq 2\beta} \frac{\sqrt{\hat{C}_j\hat{C}_k}}{2}\left[\mathrm{E}(A_jA_k) - i\mathrm{E}(A_jB_k) + i\mathrm{E}(A_kB_j) + \mathrm{E}(B_jB_k)\right] e^{-2\pi ik\tau}e^{2\pi i(j-k)t} \\
&= \sum_{|j|\leq 2\beta} \frac{\hat{C}_j}{2}\left[\mathrm{E}(A_j^2) + \mathrm{E}(B_j)^2\right] e^{-2\pi ij\tau} \\
&= \sum_{|j|\leq 2\beta} \hat{C}_j e^{2\pi ij\tau}.
\end{aligned}$$

Therefore, referencing (A.15), it follows that $\mathrm{E}(z(t)\overline{z(t+\tau)}) = C(\tau)$.