Statistical Inference and Learning for Topological Data Analysis

by

CHUL MOON

(Under the Direction of Nicole A. Lazar)

Abstract

Topological data analysis (TDA) is a rapidly developing collection of methods for studying the shape of data. Persistent homology is a prominent branch of TDA which analyzes the dynamics of topological features of a data set. We introduce statistical inference and learning methods for persistent homology of three types of data: point clouds, fingerprints, and rock images. First, we illustrate a topological inference plot for point cloud data, called the persistence terrace. The suggested plot allows robust and scale-free inference on the size and point density of topological features. Second, we suggest a new interface between persistent homology and machine learning algorithms and apply it to the problem of sorting fingerprints into pre-determined groups. We achieve near state-of-the-art classification accuracy rates by applying TDA to minutiae points and ink-roll images. Last, we present a statistical model for analysis of porous materials using persistent homology. Our model enables us to predict the geophysical properties of rocks based on their geometry and connectivity.

Index words: Persistent Homology, Statistical Inference, Machine Learning, Statistical Modeling

Statistical Inference and Learning for Topological Data Analysis

by

CHUL MOON

B.E., Pohang University of Science and Technology, Republic of Korea, 2011

B.S., Pohang University of Science and Technology, Republic of Korea, 2011

M.A., Seoul National University, Republic of Korea, 2013

M.S., University of Georgia, 2016

A Dissertation Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2018

Statistical Inference and Learning for Topological Data Analysis

by

CHUL MOON

Major Professor:	Nicole A. Lazar
Committee:	Lynne Billard
	Noah Giansiracusa
	Cheolwoo Park
	T.N. Sriram

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
August 2018

Acknowledgments

Firstly, I would like to express my sincere gratitude to my advisor Prof. Nicole A. Lazar for her continuous support, patience, and motivation during my Ph.D. study. I have learned what it is to be a devoted teacher, responsible advisor, and good statistician. I have been fortunate to have her as a mentor.

I would like to thank the rest of my thesis committee: Prof. Lynne Billard, Prof. T.N. Sriram, Prof. Cheolwoo Park, and Prof. Noah Giansiracusa for their insightful comments and encouragement. In particular, I am grateful to Prof. Cheolwoo Park, who provided me the opportunity to work as a research assistant. Furthermore, I am thankful to Prof. Noah Giansiracusa who has been a fantastic collaborator, and who has encouraged me to become an independent researcher. I would like to thank Dr. Kim Gilbert who has helped me to pursue my career in academia. My sincere thanks also go to Dr. Scott A. Mitchell who provided me an opportunity to join the wonderful project as an intern at Sandia National Laboratories.

I thank my fellow friends in the statistics department for the stimulating discussions and for all the fun we have had in the last five years. Most importantly, I am deeply thankful to Dr. Kang for continued support and encouragement.

Last but not the least, I would like to thank my parents, whose love and guidance are with me wherever I go.

This research was partially supported by National Science Foundation (NSF IIS-1607919) and by an appointment with the NSF Mathematical Sciences Summer Internship Program sponsored by the National Science Foundation, Division of Mathematical Sciences (DMS). This program is administered by the Oak Ridge Institute for Science and Education (ORISE) through an interagency agreement between the U.S. Department of Energy (DOE) and NSF. ORISE is managed by ORAU under DOE contract number DE-SC0014664.

Table of Contents

Page

Acknowledgments iv

List of Figures viii

List of Tables xiii

Chapter

1 Introduction 1

1.1 Complexes 2

1.2 Homology 4

1.3 Persistent Homology 6

2 Topological Inference Tool for Point Cloud Data: Persistence Terrace 12

2.1 Point Cloud Data 12

2.2 Two Estimation Methods 13

2.3 Persistence Terrace 17

2.4 Simulation Study 25

2.5 Discussion 31

3 Interface between Topological Data Analysis and Statistical/Machine Learning 33

3.1 Background 33

3.2 Features for Interface between TDA and SML 34

3.3 Feature Selection and Learning Methods 38

3.4 Application to Fingerprint Classification 40
3.5 Discussion 52

4 Statistical Analysis Pipeline for Porous Material Images using Topological Data Analysis 55
4.1 Background 55
4.2 Data 55
4.3 Statistical Analysis Pipeline 56
4.4 Results 67
4.5 Discussion 72

5 Conclusion and Future Research Directions 75
5.1 Improvement of Persistence Terrace as an Inference Tool 76
5.2 Interface and Fingerprint Application 76
5.3 Porous Material Analysis 79

Bibliography 81

Appendix 90

List of Figures

1.1 Scatterplot of points (upper-left), balls of diameter ϵ centered at the points (upper-right), Cěch complex (lower-left), and Vietoris-Rips complex (lower-right). Figure from Ghrist (2008) 4

1.2 Chain, cycle, and boundary groups. Figure from Zomorodian and Carlsson (2005). 5

1.3 Betti numbers of a point (left), circle (center), and torus (right). 6

1.4 Sequence of Vietoris-Rips complex for point data and the corresponding barcode plot. Figure from Ghrist (2008) 8

1.5 Scatterplot of points concentrating around two circles and the corresponding barcode plots and persistence diagrams. 9

1.6 Two persistence diagrams of purple squares and green circles (left) and the weighted graph for Wasserstein distance computation (right). Figure from Munch et al. (2015). 11

2.1 Examples of point cloud data. Points taken from the surface of a teapot from MATLAB (2016) (left) and Locations of earthquakes from Kious and Tilling (1996) (right). 12

2.2 Scatterplots of noise added data of Figure 1.5 and the corresponding barcode plots and persistence diagrams. Noise in the data creates many "false" loops. 13

2.3 Super-level sets of a heel bone structure. Modified Figure 12 of Turner et al. (2014) 14

2.4 Manifolds and Morse-based persistence diagrams at two smoothing parameters for the "two noisy circles" point cloud in Figure 2.2a. Each smoothing parameter leads to only one persistent loop; a circle is lost no matter what smoothing parameter we choose. 15

2.5 Scatterplot of three 200-point circles and a Morse persistence diagram. We label the circles and the corresponding β_1 features in the persistence diagram to illustrate the inverse relation here between feature persistence and circle radius due to point density. 16

2.6 The β_1 persistence terrace for the three circles data of Figure 2.5a. The labels in Figure 2.6b match the three terrace layers with the corresponding circles in Figure 2.5a. 18

2.7 The terrace area plot of the persistence terrace in Figure 2.6. The near-zero flattening of the graph at terrace height 4 correctly suggests that there are 3 significant topological features in the data (the height ≥ 4 layers are noise). . 19

2.8 Barcode plots at smoothing parameters 0.2, 0.6 and 1, corresponding to vertical slices of the persistence terrace. Barcode plots traditionally use the x-axis for the filtration value so we rotated counterclockwise 90 degrees to match with the y-axis of the terrace. 21

2.9 Scatterplot of the noise-added two circles data and β_1 persistence terrace. The two distinct terrace layers correctly suggest two significant topological features, which would be nearly impossible to detect using conventional Rips or Morse approaches. 22

2.10 Scatterplot, manifold, and persistence terrace of two circles data with different densities. The high density circle b smooths to a high altitude volcano and appears as a persistence terrace layer stretching along the y-axis in the satellite view (layer b). 23

2.11 Scatterplot, manifold, and persistence terrace of two circles with equal density but unequal radius data. The volcano manifold from the small circle *a* fills in more quickly as the smoothing parameter increases and appears as a persistence terrace layer stretching along a narrow stretch of the x-axis in the satellite view (layer *a*). 24

2.12 Scatterplot of four noisy shapes data and the Rips persistence diagram from which it is essentially impossible to infer precisely four significant topological features. 26

2.13 Terrace area plot and satellite view of β_1 persistence terrace for the noisy four shape data. There are four prominent layers, corresponding to the four shapes, though one of them (layer *c*) is stacked on top of another (layer *b*). 27

2.14 Scatterplot of 6,500 points sampled from a muscle tissue cross-sectional image, with added boundary lines to close the muscle fiber loops. 28

2.15 Terrace area plot and satellite view of the β_1 persistence terrace of the point cloud sampled from the muscle tissue cross-sectional image. The shape of the terrace area plot suggests at least 9–10 overlapping terrace layers, and the hand-drawn arrow in the persistence terrace reveals a non-overlapping layer, giving 10–11 total loops in the data. 30

3.1 Scatterplots of four density settings 37

3.2 Number of connected components of density data 38

3.3 Coefficients of fitted models 39

3.4 Examples of the three fingerprint classes from the dataset NIST SD-27 . . . 42

3.5 Minutiae 42

3.6 (a) Cropped images of a loop (left), whorl (middle), and arch (right). (b) Scatterplots of the corresponding normalized minutiae coordinates (the vertical axis is the orientation, so the top and bottom squares should be identified). (c) The 0-dimensional (gray) and 1-dimensional (black) barcodes for the unoriented minutiae point clouds in $\mathbb{R}^2$. (d) The barcodes for the minutiae point clouds in $\mathbb{R}^2 \times S^1$ with the metric d_2 defined earlier. Precisely interpreting these barcodes is not necessary; for the purposes of supervised learning we simply need that the barcodes reflect *some* relevant global geometric structure in the fingerprints. 44

3.7 Top row: an ink-roll JPEG, after normalizing and inverting, viewed as a 3D surface, and the 0-dimensional (gray) and 1-dimensional (black) barcodes of the superlevel sets. Middle row: the same image after slanting by the function $f(x, y) = y$, and its superlevel set barcodes. Bottom row: same slanting, but first the image is thresholded to convert from grayscale to black-and-white, and here the barcodes use sublevel sets. We employ these variants (and the others described earlier) in an effort to access as much geometry of the ridge pattern as possible. 47

4.1 Grayscale (left) and binarized (right) image slices of the Selma Chalk by Yoon and Dewers (2013). 56

4.2 Components of a cubical cell complex; 0-cell, 1-cell, 2-cell, and 3-cell, from left to right. 58

4.3 SEDT converted image (left) and sequential changes of cubical cell complexes of the images. Blue colored pixels construct the cubical complexes. Modified Figure of Robins et al. (2016). 59

4.4 Examples of persistence diagrams dimension 0, 1, and 2 59

4.5 Persistence diagram divided by 5×5 pixels (left), concatenated persistence diagram (middle) and vectorized persistence diagram (right) 62

4.6 Representation of a vectorized persistence diagram as a linear combination of principal components 65
4.7 "large n small p" vs. "large p small n" data. 66
4.8 Percentages of three regions of dimension one persistence diagram. The labels of three regions correspond to Figure 4.4. 68
4.9 Average SSIM (MSSIM_{PH}) of Selma group chalk. 69
4.10 Average SSIM (MSSIM_{PH}) of Bentheimer (top) and Doddington (bottom). . 71
4.11 Fitted vs. actual plots of porosity (first row), permeability (second row), anisotropy (third row), and tortuosity (last row) at size 150^3 (first column), 300^3 (second column), and 400^3 (last column). Points closer to the 45 degree line imply more accurate prediction results. 73
1 Resolution of the persistence terrace according to the number of smoothing parameters. With an increase in the number of smoothing parameters, the resolution increases, although the general picture stays the same. 92
2 Examples of pore and grain structures in the corresponding dimension zero (top row), one (second row), and two (last row) persistence diagrams. 93
3 Dimension 0 persistence diagrams under nine stress-strain levels. 94
4 Dimension 1 persistence diagrams under nine stress-strain levels. 95
5 Dimension 2 persistence diagrams under nine stress-strain levels. 96
6 Average SSIM (MSSIM_{PH}) of Rotleigend. 97

List of Tables

1.1 Interpretation of Betti numbers of dimension zero, one and two (β_0, β_1 and β_2). 6

2.1 Comparison of direct and robust estimation approaches 17

3.1 List of possible functions and variables 35

3.2 The peak accuracy rates obtained when selecting various sets of features, removing highly correlated ones, then performing backwards elimination on the remaining ones. The rate listed is the maximum obtained this way for each group, and the number of features is the size of the subset(s) of features achieving this rate. 50

3.3 The confusion matrices for the two best classifiers using 32 features selected from 552 features (top) and 1-dimensional features (bottom). 51

3.4 With the notation for our features introduced in Section 3.4.3, these are the 32 features selected from among all 552 that achieve our best rate, 93.1%. . . 52

4.1 Data summary 56

4.2 Interpretation of persistence diagrams 60

Chapter 1

Introduction

A defining characteristic of many modern data applications – whether they fall under the heading of "Big Data" or not – is their unstructured nature. It can no longer be assumed that data will come to us for analysis in regular arrays with fixed numbers of rows and columns and a single observation in each cell, usually representing the measured value of a particular variable for a particular unit. Similarly, questions of scientific interest have been shifting in recent years. In settings such as neuroimaging and genetics, to give two prominent examples, researchers are focusing on questions about network structure, interactions between brain regions or regions on the genome, and the like. Such questions are not amenable to traditional statistical procedures based on simple array-structured data. Accordingly, recent years have seen the development of methods for functional (Ramsay and Silverman, 2005), object-oriented (Marron and Alonso, 2014), and symbolic (Billard and Diday, 2007) data. All of these aim to tackle situations where the basic unit of analysis is something other than a traditional observation; rather, it can now be an entire image, or a histogram, or a function.

A relatively new, emergent approach is topological data analysis (TDA), which focuses on the "shape" or "structure" of a data set (Carlsson, 2009; Lum et al., 2013). TDA provides quantitative methods to analyze the underlying geometric and topological structures of data. TDA has been successfully applied to various areas; examples include the study of brain artery structure (Bendich et al., 2016); brain network (Lee et al., 2012; Petri et al., 2014); protein structure (Gameiro et al., 2015); granular packing (Saadatfar et al., 2017); viral genomics (Chan et al., 2013); porous material (Robins et al., 2016); dynamical system (Adams et al., 2017); bone structure (Heo et al., 2012; Turner et al., 2014); and breast cancer

(Nicolau et al., 2011). However, a statistical approach, which takes into account randomness and noise of data, has started to be considered very recently (Chazal and Michel, 2017). A major challenge in the TDA approach is to decide which features are real, in the sense of representing meaningful structure in the data, and which are artifacts of the noise inherent in all measured data. This is a question of statistical inference, and as such, recent years have seen the development of statistical methods to blend with the topological concepts underlying TDA.

In this dissertation, we propose statistical learning methods and data analysis pipelines for topological inference using a prominent branch of TDA called persistent homology. We also establish our approaches by presenting applications to various types of data. The rest of this dissertation is organized as follows. In the rest of Chapter 1, we summarize the theoretical background of (persistent) homology (Hatcher, 2002; Edelsbrunner and Harer, 2008, 2010; Ghrist, 2008; Zomorodian and Carlsson, 2005; Zomorodian, 2012). Chapter 2 introduces a summary plot for point cloud data called the persistence terrace and demonstrates its use with simulated and real data. In Chapter 3 we explore the interface between topological data analysis and machine learning as applied to the fingerprint classification problem. Chapter 4 explains a statistical pipeline for porous material images using persistent homology and presents analysis results. Finally, in Chapter 5, we summarize our main findings and propose directions for future work.

1.1 Complexes

A topological space is a mathematical object abstracting the intuitive notions of nearness, connectivity, and continuity. In order to perform computations on a topological space, one needs to discretize and represent the space with a finite amount of information. Fortunately, this is possible for most spaces appearing in practice through the process of triangulation: we encode the space as a *simplicial complex*, which means a collection of vertices (0-simplex),

edges (1-simplex), triangles (2-simplex), tetrahedra (3-simplex), and higher-dimensional simplices, together with the data of how all these pieces are attached. In the construction of the simplicial complex, simplexes are glued together in a way that two adjacent simplexes share their faces. For instance, the circle is a single vertex with a single edge attached at both ends to the vertex. There exist different ways to build the simplicial complex and we will explain two approaches: Cĕch and Vietoris-Rips complexes.

Let X be a set of points in a metric space. For a given filtration value ϵ, we draw balls of radius $\epsilon/2$ centered at the data points. First, a *Cĕch complex* is defined by the intersection between those balls. If there are n balls that share the intersection region, then the corresponding n data points form an $n-1$-simplex. The Cĕch complex at the filtration value ϵ is a union of those simplexes and is represented as $\mathcal{C}(X, \epsilon)$. The Cĕch complex has the property called the Cĕch theorem or the nerve theorem (Borsuk, 1948). The theorem states that if we sample enough data from a topological space T and construct the Cĕch complex, then the complex reflects the topology of T. Thus, the Cĕch complex can model the topology of a point cloud data. However, a drawback of the Cĕch complex is that it is computationally expensive.

A *Vietoris-Rips (Rips) complex* is a good approximation of the Cĕch complex. The Rips complex requires less computational load than the Cĕch complex because it is determined by only pairwise distances between points. Let's assume that we have the same point cloud data X. For given ϵ, m data points whose pairwise distances are smaller than ϵ form a $(m-1)$-simplex. The Rips complex of X given ϵ is a union of the simplexes and expressed as $\mathcal{R}(X, \epsilon)$.

The cost of constructing the Rips complex instead of the Cĕch complex exist: it is hard to guarantee the Cĕch theorem for the Rips complex. The Rips complex might not fully demonstrate the topology of the corresponding space. However, the Rips complex still carries topological properties of point cloud data. For all $\epsilon > 0$, the following relationships hold:

$$\mathcal{C}(X, \epsilon) \subset \mathcal{R}(X, \epsilon) \subset \mathcal{C}(X, 2\epsilon).$$

Although the Rips complex $\mathcal{R}(X, \epsilon)$ does not satisfy the nerve theorem, it contains topological information of the Cĕch complexes somewhere between $\mathcal{C}(X, \epsilon)$ and $\mathcal{C}(X, 2\epsilon)$. Therefore, the Rips complex can be a good alternative to the Cĕch complex.

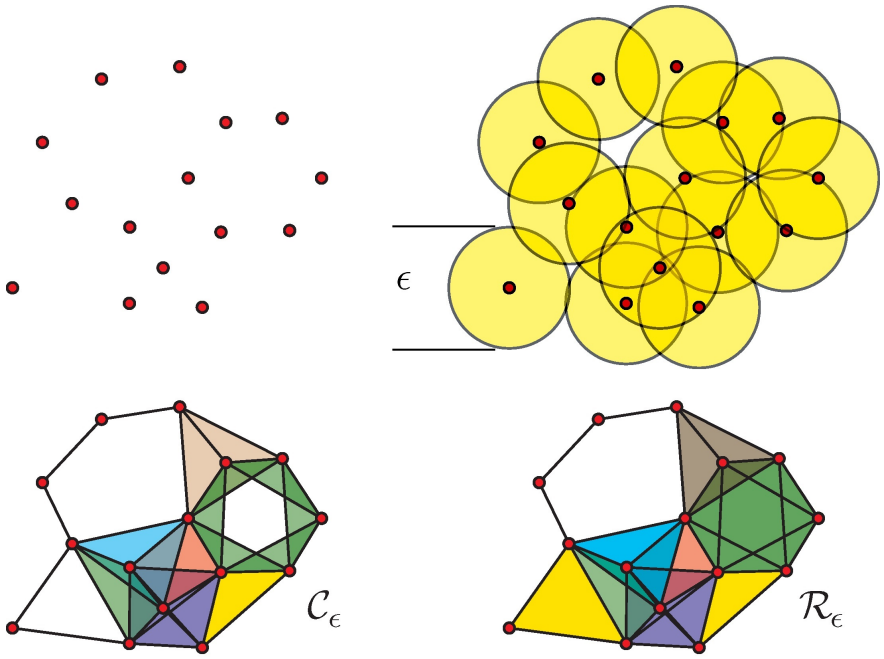


Figure 1.1: Scatterplot of points (upper-left), balls of diameter ϵ centered at the points (upper-right), Cĕch complex (lower-left), and Vietoris-Rips complex (lower-right). Figure from Ghrist (2008)

Figure 1.1 illustrates the construction of the Cĕch complex and Rips complex for point cloud data of 15 data points. The difference between the two complexes can be seen in the green region on the lower right (six points). The six points construct a set of multiple 2-simplexes (triangles) in the Cĕch complex. On the other hand, the same points construct a 5-simplex in the Rips complex because pairwise distances are smaller than ϵ.

1.2 Homology

Any metric space (such as $\mathbb{R}^n$), or a subset thereof, can be viewed as a topological space. Homology is used to quantify the topological characteristics of a topological space. Let K be a simplicial complex such as $\mathcal{C}(X, \epsilon)$ and $\mathcal{R}(X, \epsilon)$. An *orientation* of a n-simplex

$\sigma = \{v_0, v_1, \cdots, v_n\}$, where v's are vertices of K, is an equivalence class of orderings of vertices of σ. An *oriented simplex* is denoted as $[\sigma]$. The nth *chain group* $\mathsf{C}_n(K)$ is the free Abelian group on K's oriented n-simplices. An element of $\mathsf{C}_n(K)$ is called the *n-chain* $c = \sum_i c_i [\sigma_i]$. The *boundary operator* ∂_n is a homomorphism defined on an oriented simplex in the n-chain c:

$$\partial_n [v_0, \cdots, v_n] = \sum_i (-1)^i [v_0, \cdots, \hat{v}_i, \cdots, v_n],$$

where $\hat{v}_i$ indicates that v_i is not included in the sequence. The boundary operator connects the chain groups as a chain complex C_* as

$$\cdots \to \mathsf{C}_{n+1} \xrightarrow{\partial_{n+1}} \mathsf{C}_n \xrightarrow{\partial_n} \mathsf{C}_{n-1} \to \cdots .$$

We can define subgroups of *kernel* and *image* using the boundary operator. If $c \in \mathsf{C}_n$ and $\partial_n(c) = 0$, then c is called a *cycle*. The set of cycles in C_n is the kernel of ∂_n, denoted as $\mathsf{Z}_n = \ker(\partial_n)$. Also, B_n is the boundary group $\mathsf{B}_n = \mathrm{im}(\partial_{n+1})$. These subgroups are nested as $\mathsf{B}_n \subset \mathsf{Z}_n \subset \mathsf{C}_n$ and the relationships are presented in Figure 1.2. The n^{th} *homology group*

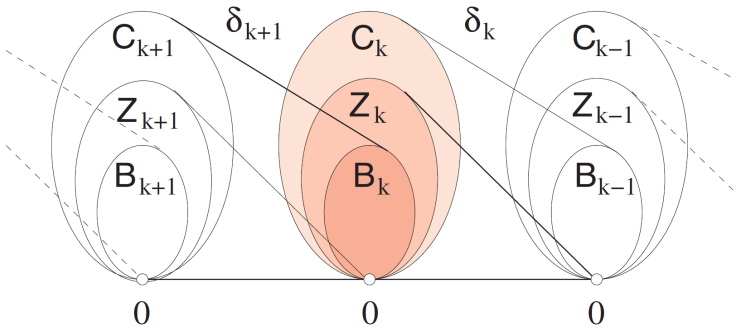


Figure 1.2: Chain, cycle, and boundary groups. Figure from Zomorodian and Carlsson (2005).

H_n is defined using a quotient group

$$H_n = \ker(\partial_n)/\mathrm{im}(\partial_{n+1}) = \mathsf{Z}_n/\mathsf{B}_n.$$

The rank of H_n is called the n^{th} *Betti number*, $\beta_n = \mathrm{rank}(H_n) = \mathrm{rank}(\mathsf{Z}_n) - \mathrm{rank}(\mathsf{B}_n)$. Hence, the n^{th} Betti number represents the number of n-dimensional holes of the topological space.

Table 1.1: Interpretation of Betti numbers of dimension zero, one and two (β_0, β_1 and β_2).

Symbol	Dimension	Counts
β_0	0	Number of connected components
β_1	1	Number of loops
β_2	2	Number of enclosed voids

Table 1.1 gives geometric interpretations of the $n = 0, 1, 2$ Betti numbers. Betti numbers are the numerical summary of homological characteristics of a topological space.

Figure 1.3 illustrates a point, circle, and torus (hollow donut) with their Betti numbers. Each has a single connected component; the circle has a single loop; the torus has two loops (in the vertical and horizontal directions) and encloses a single solid void. Topological characteristics of the three different objects are summarized with three-dimensional Betti numbers.

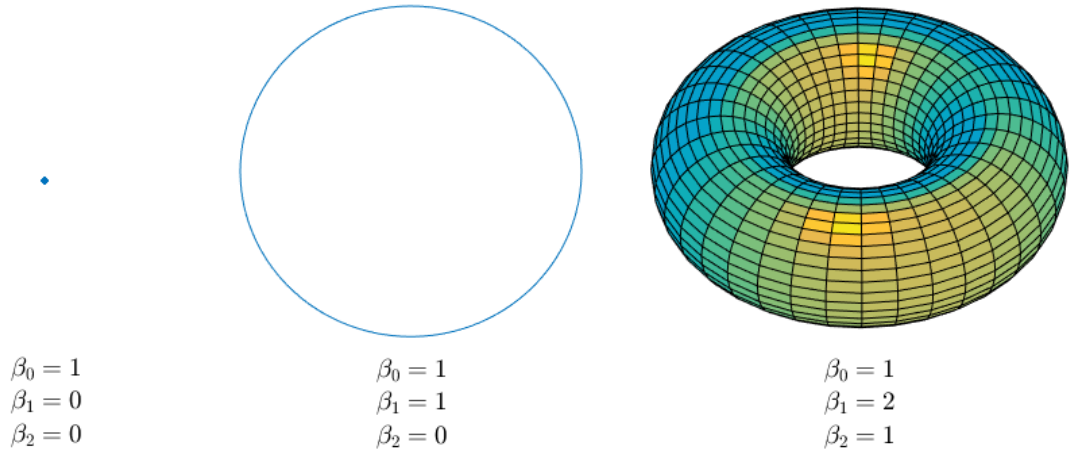


Figure 1.3: Betti numbers of a point (left), circle (center), and torus (right).

1.3 Persistent Homology

For a 1-parameter family of topological spaces, or simplicial complexes, *persistent homology* offers a method of quantifying the dynamics of topological features (e.g., when holes appear

and disappear). A 1-parameter family of simplicial complexes where simplices are added but never removed is called a *filtered* simplicial complex

$$\emptyset = K^0 \subseteq K^1 \subseteq \cdots \subseteq K^m = K.$$

The parameter is then usually called a *filtration* parameter. Let us assume that a filtration is defined to be a radius of a ball as illustrated in Section 1.1. Then, as the filtration value increases, the size of balls of diameter ϵ gets bigger. Therefore, the number of data points whose balls share the intersection region (the Cĕch complex) or whose pairwise distances are smaller than ϵ (the Rips complex) increases. These data points form higher dimensional simplices and are added to the filtered simplicial complex. Both Cĕch and Rips complexes have the inclusion relationship such that

$$K^1 = \mathcal{R}(X, \epsilon_1) \subseteq K^2 = \mathcal{R}(X, \epsilon_2),$$

for $\epsilon_1 \leq \epsilon_2$. Figure 1.4 illustrates the sequential changes of the Rips complex.

For $i \leq j$ and filtered simplicial complexes $K^i \subseteq K^j$, the n^{th} (i, j)-*persistent homology* $\iota_n(i, j)$ is defined to be the image of the induced homomorphism between the homology groups,

$$\iota_n(i, j) : H_n(K^i) \to H_n(K^j).$$

The ranks of the images of $\iota_n(i, j)$, $\beta_n^{i,j} = \text{rank}(\text{im}(\iota_n(i, j)))$, are called the *persistent Betti numbers* of ι. We may define the persistent homology group of K^i in a different way:

$$H_n^{i,j} = \mathsf{Z}_n^i / \left(\mathsf{B}_n^{i+j} \cap \mathsf{Z}_n^i \right),$$

where H_n^i, Z_n^i, and B_n^i are the groups associated with K^i and a boundary operator ∂_n^i.

The *barcode* records the "birth" (appear) and "death" (disappear) of a specific n-dimensional hole over the filtration. As the filtration varies, the n-simplex either creates a n-dimensional hole or destroys a $n-1$-dimensional hole. Each topological feature can be matched with its creator and destroyer. The lifetime of a homology class is given in the form of interval data of [birth filtration value, death filtration value]. The relationship between

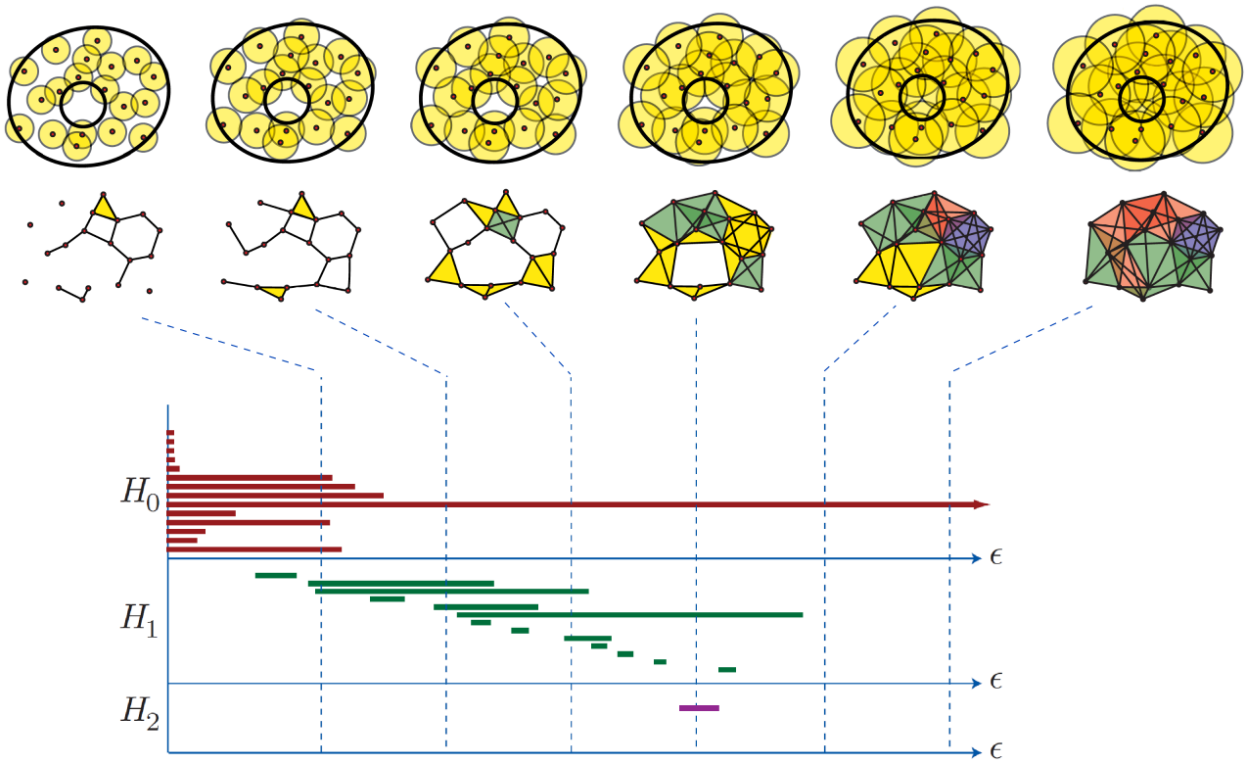


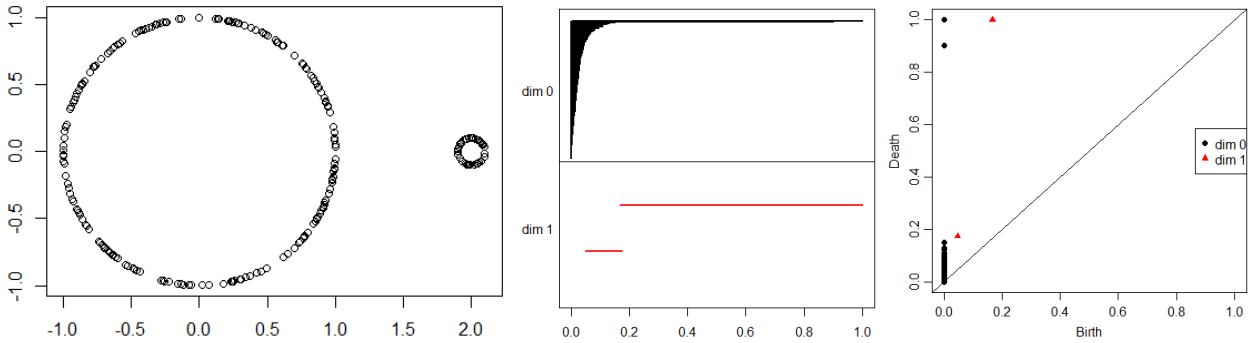
Figure 1.4: Sequence of Vietoris-Rips complex for point data and the corresponding barcode plot. Figure from Ghrist (2008)

the Betti number and the barcode can be easily understood with the popular two economics terms *stocks* and *flows*. The Betti number is a stock variable that counts the number of n-dimensional holes of the filtered simplicial complex at a certain filtration value. It does not report how topological features persist over the filtration range. On the other hand, the barcodes work as a flow variable that describes the persistence of the topological feature. The n^{th} Betti number of $\mathbb{R}(X, \epsilon_i)$ is the same as the total number of n-dimensional barcodes standing at ϵ_i.

1.3.1 Summary Plots

Persistent homology can be visually summarized with the *barcode plot*: for each dimension k, plot a collection of horizontal intervals whose left endpoint is the filtration value at which a

particular k-dimensional homology class is born and whose right endpoint indicates its death. The number of intervals over a filtration value is the Betti number β_k at that value. A more compact visual summary is the *persistence diagram*: here each homology class is plotted as a point whose x-coordinate is the birth time and y-coordinate is the death time; different symbols are used to distinguish homological dimension. The prominence, or persistence, of a topological feature corresponds to the length of a bar in a barcode plot, or to the height above the diagonal line $y = x$ in a persistence diagram.



(a) Scatterplot (b) Barcode plot (c) Persistence diagram

Figure 1.5: Scatterplot of points concentrating around two circles and the corresponding barcode plots and persistence diagrams.

Figure 1.5 shows the scatterplot, barcode plot, and persistence diagram obtained using the Rips complex. There are two long lines on dimension 0 and 1 barcode plots and two points of dimension 0 and 1 far from 45-degree line on persistence diagrams. From the both summary plots, we can see that two significant connected components and two loops exist.

1.3.2 Comparison of Results

As presented in the previous subsection, persistence can be encoded in barcodes, a multiset of intervals. If noise or measurement errors on data significantly affect barcodes, then features might not be represented well. Therefore, stability of the persistence information under perturbations is important for making inference. This topic has been studied for the persistence diagrams. Let D_X and D_Y be the persistence diagrams of metric space X and Y. For given metrics d_1 on persistence diagrams and d_2 on the metric spaces, the stability

of persistence diagrams holds when

$$d_1(D_X, D_Y) \leq d_2(X, Y).$$

If the persistence diagram is stable, then small changes in metric spaces (measured as $d_2(X, Y)$) induce small changes in persistence diagrams (measured as $d_1(D_X, D_Y)$). Therefore, persistence diagrams can be used to measure geometric or topological differences between spaces.

The persistence diagram distances measure the dissimilarity between diagrams. The persistence diagram distance needs to compare diagrams that include different numbers of points. This is because the number of barcodes created varies from dataset to dataset. Also, the distances need to satisfy an important property called stability under perturbations. In this subsection, we illustrate three persistence diagram distances; Wasserstein, bottleneck, and persistent landscape distances.

Let $x_i = (a_i, b_i)$, $y_j = (a_j, b_j)$, and $d(x_i, y_j) = ||x_i - y_j||_\infty = \max(|b_i - b_j|, |a_i - a_j|)$. Bottleneck distance between persistence diagrams D_X and D_Y is defined as

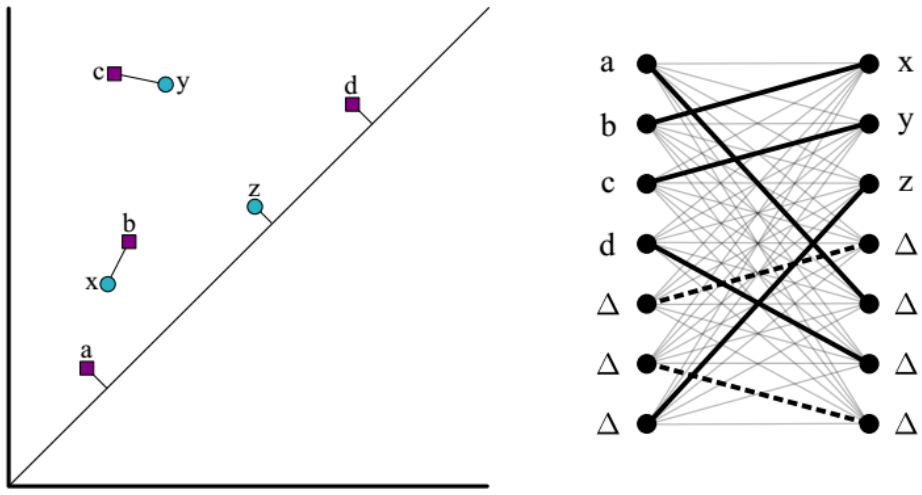
$$W_\infty(D_X, D_Y) = \inf_\gamma \sup_{x \in D_X} d(x, \gamma(x)) = \inf_\gamma \sup_{x \in D_X} ||x - \gamma(x)||_\infty$$

where γ is the bijections from D_X to D_Y. Bottleneck distance between two persistence diagrams has an upper bound by L_∞-distance. The stability of the bottleneck distance is shown in Cohen-Steiner et al. (2007) and Chazal et al. (2012).

p-Wasserstein distance between D_X and D_Y is given by

$$W_p(D_X, D_Y) = \left(\inf_\gamma \sum_{x \in D_X} d(x, \gamma(x))^p \right)^{\frac{1}{p}} = \left(\inf_\gamma \sum_{x \in D_X} ||x - \gamma(x)||_\infty^p \right)^{\frac{1}{p}}$$

Figure 1.6 shows the computation of the Wasserstein distance. Two persistence diagrams are presented in Figure 1.6a with purple squares and green circles. The Wasserstein distance minimizes the cost of coupling on the weighted graph in Figure 1.6b. The chosen couplings are shown as lines in Figure 1.6a and bold lines in Figure 1.6b. Wasserstein distance's stability conditions and results are shown in Cohen-Steiner et al. (2010).



(a) Two persistence diagrams (b) Weighted graph

Figure 1.6: Two persistence diagrams of purple squares and green circles (left) and the weighted graph for Wasserstein distance computation (right). Figure from Munch et al. (2015).

Bubenik (2015) defines the landscape distance on the persistence landscape. Let M be the persistence module. A rank function λ is defined as

$$\lambda(a,b) = \begin{cases} \beta^{a,b} & \text{if } a \leq b \\ 0 & \text{otherwise,} \end{cases}$$

where $\beta^{a,b} = \dim(\text{im}(M(a \leq b)))$. The persistent landscape is a sequence of functions $\lambda_k(t) = \lambda(k,t)$, where

$$\lambda_k(t) = \sup(m \geq 0 | \beta^{t-m,t+m} \geq k)$$

for $m = (a+b)/2$. Let λ^X and λ^Y be the persistence landscapes of persistence diagrams D_X and D_Y. Then, the p-landscape distance is defined as

$$\Lambda_p(D_X, D_Y) = ||\lambda^X - \lambda^Y||_p,$$

where $||\lambda||_p = \sum_{k=1}^{\infty} ||\lambda_k||_p$. The stability of the persistent landscape distance is proven in Bubenik (2015).

Chapter 2

Topological Inference Tool for Point Cloud Data: Persistence Terrace

2.1 Point Cloud Data

A *point cloud*—a finite, unordered collection of points in $\mathbb{R}^n$ or some other metric space—is uninteresting as a topological space since it is discrete: β_0 is the number of points in the cloud and $\beta_k = 0$ for all $k \geq 1$. Such data commonly arise through discrete sampling of continuous objects, the locations of events, and structure of crystals, etc. Figure 2.1 shows examples of point cloud data.

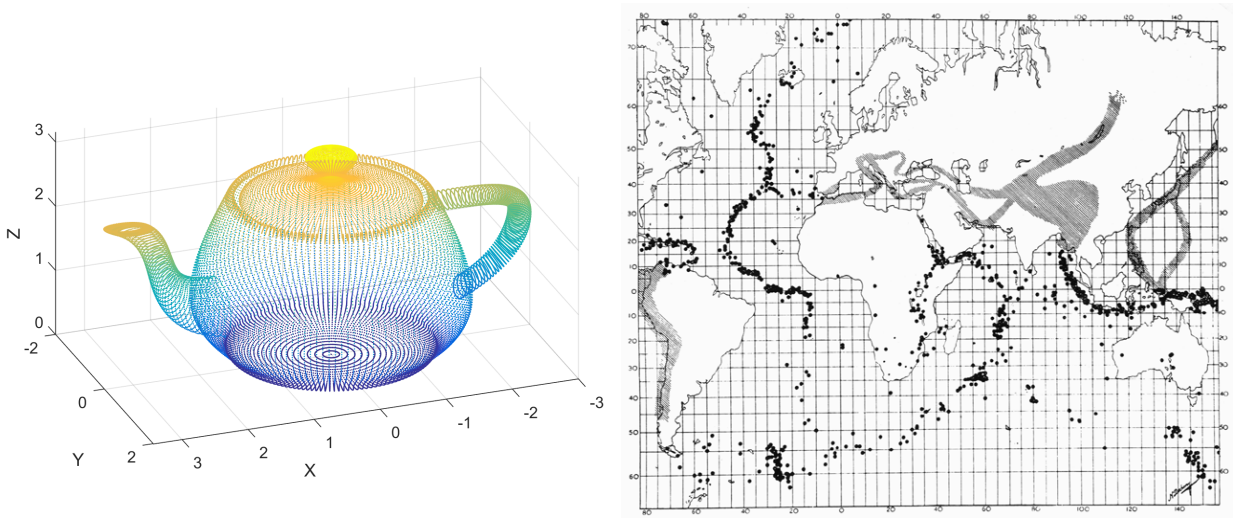


Figure 2.1: Examples of point cloud data. Points taken from the surface of a teapot from MATLAB (2016) (left) and Locations of earthquakes from Kious and Tilling (1996) (right).

The goal of point cloud data analysis is to reconstruct the objects/processes producing the point clouds and/or to compare structural properties of the point clouds themselves. TDA introduces methods of building a filtered simplicial complex from a point cloud, thereby reconstructing multi-scale topological features of the object or distribution from which the

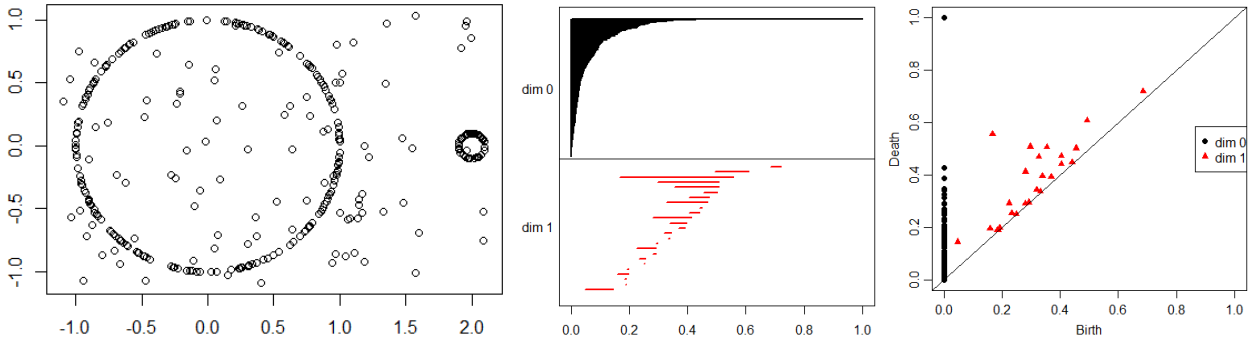
points are sampled. The standard approach is the Rips complex: points in the cloud serve as vertices; pairs of points are joined by an edge when their distance is less than the filtration value.

2.2 Two Estimation Methods

2.2.1 Direct Estimation

Point Cloud Data → Nested Complexes → Persistence Diagram or Barcode Plot

We denote *direct estimation* as an approach to construct complexes directly from point cloud data and compute persistent homology. The persistence of a topological feature is heavily affected by its size. For example, Figure 1.5 summarizes persistent homology computation result by direct estimation. The longer dimension one line (red) in Figure 1.5b and the red triangle on the top in Figure1.5c correspond to the large circle in Figure 1.5a. The persistence of the dimension one feature is proportional to the radius of the circle it comes from.



(a) Scatterplot (b) Barcode plot (c) Persistence diagram

Figure 2.2: Scatterplots of noise added data of Figure 1.5 and the corresponding barcode plots and persistence diagrams. Noise in the data creates many "false" loops.

Figure 2.2 shows the noise-added scatterplot, barcode plot, and persistence diagram obtained using the Rips complex. When noise is added, many false loops appear, and it becomes difficult to infer the actual number of circles in the point cloud.

2.2.2 Robust Estimation

Point Cloud Data → Manifold → Level Sets → Persistence Diagram or Barcode Plot

To overcome sensitivity to noise and outliers, *robust* TDA approaches have been developed using distance to a measure (Chazal et al., 2011, 2017), kernel distances (Phillips et al., 2013), kernel density estimators (Fasy et al., 2014), and kernel estimations (Bobrowski et al., 2017). These methods first transform a discrete point cloud into a continuous *manifold* via a smoothing function.

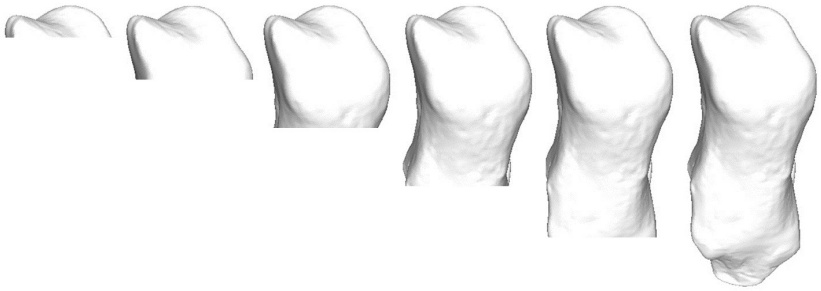
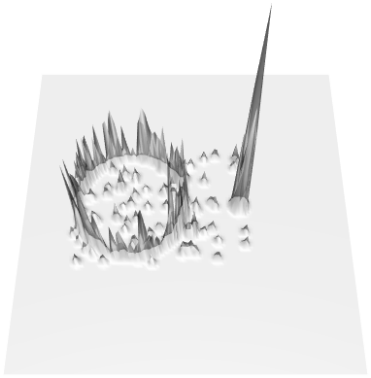


Figure 2.3: Super-level sets of a heel bone structure. Modified Figure 12 of Turner et al. (2014)

Let $\mathbb{M}$ be a manifold and $f : \mathbb{M} \to \mathbb{R}$ be a function. A function f can be seen as a height function in a certain direction. A level set $\mathbb{M}_f$ is the subset of the manifold in the direction of f. For example, a super-level set is defined as $\mathbb{M}_{f\geq a} = \{x \in \mathbb{M} \mid f(x) \geq a\} = f^{-1}([a, \infty))$. If a is the minimum value of the function, then $\mathbb{M} = \mathbb{M}_{f\geq a}$. Figure 2.3 shows super-level sets of a heel bone. In this example, f is the height of the bone in the vertical direction. Topological characteristic of a manifold can be revealed if the function is appropriately defined. Let M be the nested manifolds $M : \mathbb{M}_{f\geq a_1} \subseteq \mathbb{M}_{f\geq a_2} \subseteq \cdots \subseteq \mathbb{M}_{f\geq a_n}$, where $a_1 \geq a_2 \geq \cdots \geq a_n$. The image of a homomorphism $f_k^{i,j} : H_k(\mathbb{M}_{f\geq a_i}) \to H_k(\mathbb{M}_{f\geq a_j})$ is the persistent homology. We can analyze the topological characteristics by computing Betti numbers and barcodes using the persistent homology.

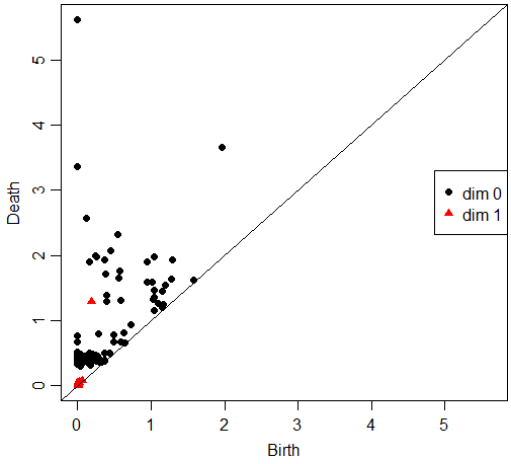
For example, graphing the mth nearest neighbor function turns a point cloud in $\mathbb{R}^2$ into a surface in $\mathbb{R}^3$; the larger the value of m, the more rounded this surface will be. The manifold

thus obtained can be filtered by its super-level sets, sets above a threshold value. The filtered super-level sets yield a 1-parameter family of topological spaces—which, when triangulated for computational purposes, is a filtered simplicial complex called the *Morse* complex of the point cloud—so once again persistent homology can be computed. The Morse complex is constructed with the smoothing function; the resulting persistent homology is more robust than that of the Rips complex. In the Morse-based approach, choosing the appropriate smoothing parameter is important for topological inference. Chazal et al. (2017) suggest a method for choosing the optimal smoothing parameter using information measures.

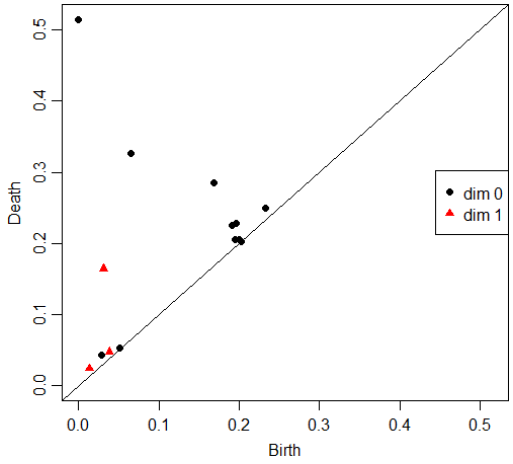


(a) Manifold at bandwidth 0.03

(b) Manifold at bandwidth 0.2



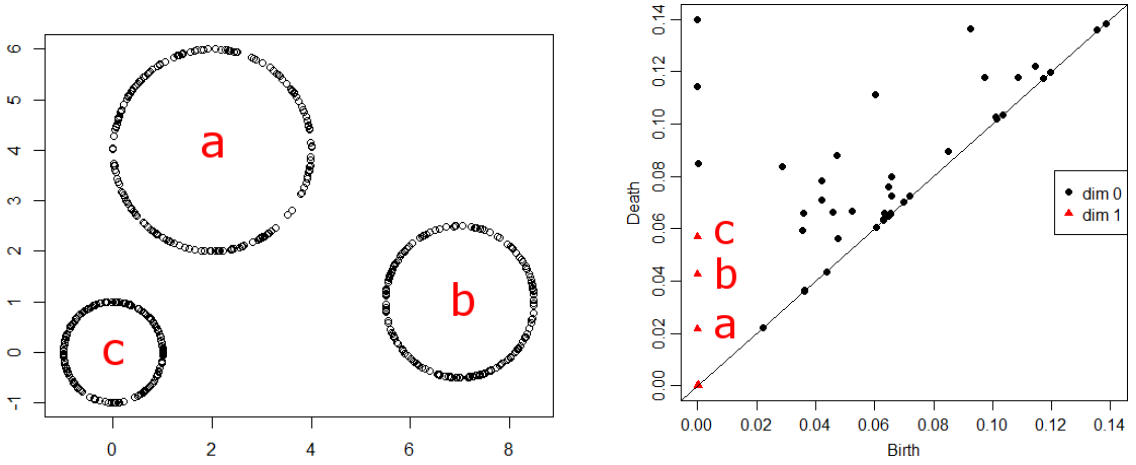
(c) Persistence diagram at bandwidth 0.03



(d) Persistence diagram at bandwidth 0.2

Figure 2.4: Manifolds and Morse-based persistence diagrams at two smoothing parameters for the "two noisy circles" point cloud in Figure 2.2a. Each smoothing parameter leads to only one persistent loop; a circle is lost no matter what smoothing parameter we choose.

There are disadvantages to these robust Morse-based smoothing approaches. First, there might not exist a single optimal smoothing parameter such that the corresponding persistent homology reveals features occurring at different scales. For example, in Figure 2.4 we use the "two noisy circles" data from Figure 2.2a and apply the kernel density estimator with two different bandwidth values. From the persistence diagrams, we see that the noise has been cleaned up by switching from the Rips to the Morse complex—but for both smoothing parameter values, one of the two circles has been washed away in the process. A second issue is that the Morse filtration only computes the number of k-dimensional holes of the level sets, not their sizes, so important scale information is lost. Figure 2.5 shows a point cloud with three circles, each containing 200 points, and its persistence diagram using kernel density estimator with bandwidth 0.2. The Morse persistence diagram indicates three prominent loops in the data, but the height of each triangle above the diagonal line (the "persistence") positively relates to the density of points on the corresponding circle and thus is negatively related to the radius.



(a) Scatterplot (b) Persistence diagram of Morse filtration

Figure 2.5: Scatterplot of three 200-point circles and a Morse persistence diagram. We label the circles and the corresponding β_1 features in the persistence diagram to illustrate the inverse relation here between feature persistence and circle radius due to point density.

2.2.3 Comparisons of Two Estimation Methods

The significance of a topological feature, in general, is recognized by the height-above-diagonal of a point in the persistence diagram. Despite important steps toward a statistical theory of quantifying significance and producing confidence interval type analysis for persistent homology (Fasy et al., 2014; Chazal et al., 2015), serious conceptual challenges remain.

In direct estimation, a small feature with high density is usually seen as insignificant because the Cĕch or Rips complex fills in the hole so quickly. On the other hand, for the robust Morse-based approach, the significance depends, in addition to height-above-diagonal, on the smoothing parameter. The smoothing parameter is chosen based on the size of data features one aims to uncover, but then height-above-diagonal in the persistence diagram reflects the density of points more than their significance (recall Figure 2.5). Therefore, whether using direct or robust estimation, it is impossible to fully capture significance with a single diagram: the significance of a feature depends on both its *size* and *density*. Table 2.1 summarizes the characteristics of the two estimation methods.

Table 2.1: Comparison of direct and robust estimation approaches

	Direct Estimation	Robust Estimation
Pros	Fast and simple	Robust
Cons	Sensitive	Smoothing parameter selection
Significance	Size	Point density

In the following section, we introduce the persistence terrace which is robust to noise and simultaneously reveals significance with regard to both size and point density of each topological feature.

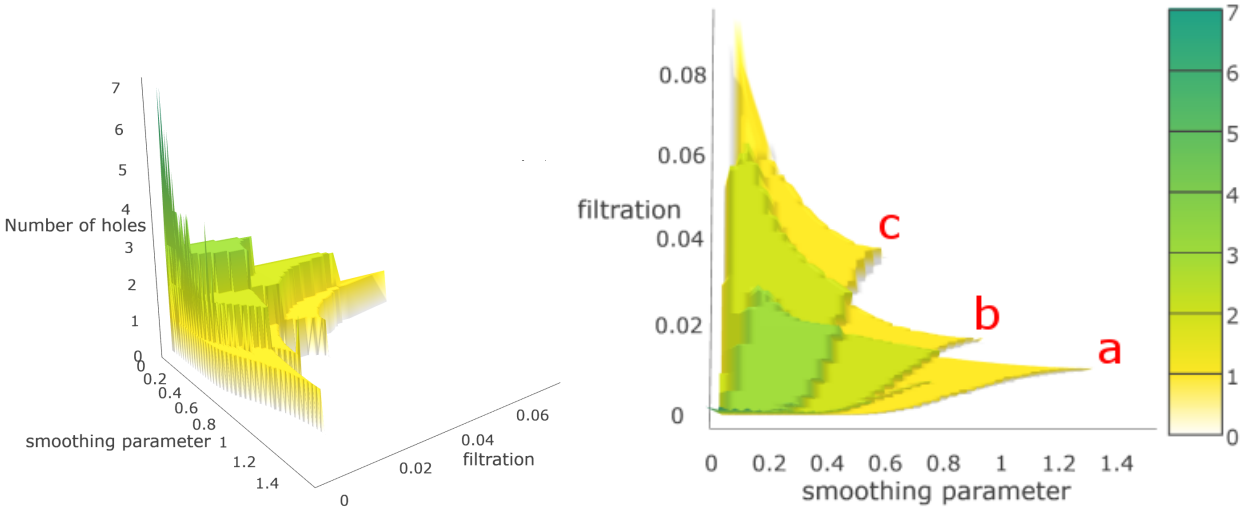
2.3 Persistence Terrace

The persistence terrace uses robust Morse-based persistent homology but incorporates a range of smoothing parameters instead of a single optimal value, similar to a scale space

analysis (Chaudhuri and Marron, 1999, 2000). For each dimension k, we plot a surface where the x-axis is the smoothing parameter, the y-axis is the filtration value, and the z-axis is the Betti number β_k. Thus, for a point cloud in $\mathbb{R}^n$, a point on the persistence terrace with coordinates (x_0, y_0, z_0) means there are z_0 holes of dimension k on the Euclidean subset

$$\{\vec{v} \in \mathbb{R}^n \mid f_{x_0}(\vec{v}) \geq y_0\},$$

where $f_{x_0} : \mathbb{R}^n \to \mathbb{R}$ is the chosen smoothing function corresponding to parameter x_0. In this analysis, we use a Gaussian kernel density estimator as the smoothing function and the bandwidth becomes the smoothing parameter. Note that this subset is topologically equivalent to its graph in $\mathbb{R}^{n+1}$ under the function f_{x_0}, which is the super-level set used in Morse-based persistent homology.



(a) Overall view (b) Satellite view

Figure 2.6: The β_1 persistence terrace for the three circles data of Figure 2.5a. The labels in Figure 2.6b match the three terrace layers with the corresponding circles in Figure 2.5a.

Figure 2.6 shows the β_1 persistence terrace of the three circles point cloud from Figure 2.5a using 50 bandwidth values from 0.01 to 1.5. Because Betti numbers are always integers, and they tend not to have gaps when the two parameters (smoothing and filtration) vary by small amounts, the surfaces plotted in a persistence terrace consist of flat layers that somewhat resemble a rice terrace (see Figure 2.6a). Each k-dimensional topological feature in the point

cloud contributes a layer to the persistence terrace; when there is a range of parameters for which multiple features are detectable, the corresponding layers in the terrace stack on top of each other and result in a higher altitude layer over their intersection.

The topological features in the point cloud are represented as terrace layers in the persistence terrace and the shape and location of the layers reflect the size and density of the corresponding features. In the satellite view of a persistence terrace, the horizontal width (x-axis direction) is positively related to the size of the feature and the vertical length of a terrace layer (y-axis direction) is proportional to the point density. Thus, we can match the topological features to the terrace regions according to their size and the point density. For example, the large-sized but low density circle a in Figure 2.5a is represented as the long horizontal width but short vertical length terrace region (layer a) in Figure 2.6b.

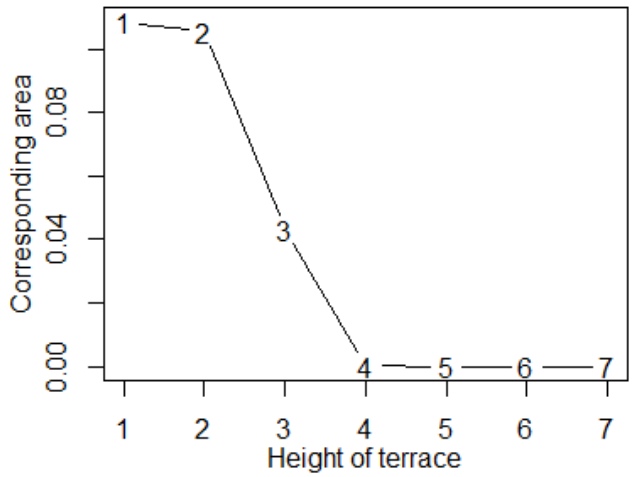


Figure 2.7: The terrace area plot of the persistence terrace in Figure 2.6. The near-zero flattening of the graph at terrace height 4 correctly suggests that there are 3 significant topological features in the data (the height ≥ 4 layers are noise).

When the number of topological features is large it can be difficult to separate distinct terrace layers by eye. We suggest a *terrace area plot* to aid in determining the significant features. The terrace area plot presents the total area of each height in the persistence terrace. Because the scales of the filtration and smoothing parameter vary from dataset to dataset, we normalize each axis so that the total area is one. The persistence terrace in Figure 2.6 has the terrace area plot seen in Figure 2.7; we see that the layers of height greater than or equal

to four are all quite small, indicating that they correspond to spikes near the origin resulting from noise, so the terrace area plot helps confirm that there are three significant topological features. Thus the terrace area plot can be used to select significant layers similar to a scree plot for a principal component analysis (though the areas in the terrace area plot can increase unlike the variances in the scree plot). The largest height among the selected layers, three in Figure 2.7, is the minimum number of significant features for which the corresponding terrace layers simultaneously overlap. If one can find layers that do not overlap with the selected height layer, then they can be counted as additional significant features, though in practice this can be tricky.

The relationship between the persistence terrace and barcodes is illustrated in Figure 2.8. Fixing a value of the smoothing parameter corresponds to taking a vertical slice of the persistence terrace. While this slice of the terrace shows the Betti number β_k at each filtration value, the barcode indicates this Betti number with β_k separate horizontal intervals. In Figure 2.8 we have chosen three different values of the smoothing parameter (0.2, 0.6 and 1) at which to slice the terrace and draw the corresponding barcode.

The persistence terrace allows us to separate out topological features even when there is no value of the smoothing parameter that could detect all features. Recall that in Figure 2.4 we try two different values of the smoothing parameter and in each case lost the information of one of the two circles (the low density large circle or the high density small circle). Nonetheless, the persistence terrace in Figure 2.9b shows two clear, distinct height-one layers; any optimal range of smoothing parameter is so small it is essentially impossible to locate. Thus, one readily infers two distinct loops in the data, one large and the other smaller and denser—even though the standard Rips approach fails due to noise and the robust Morse approaches fail due to inadequacies of smoothing.

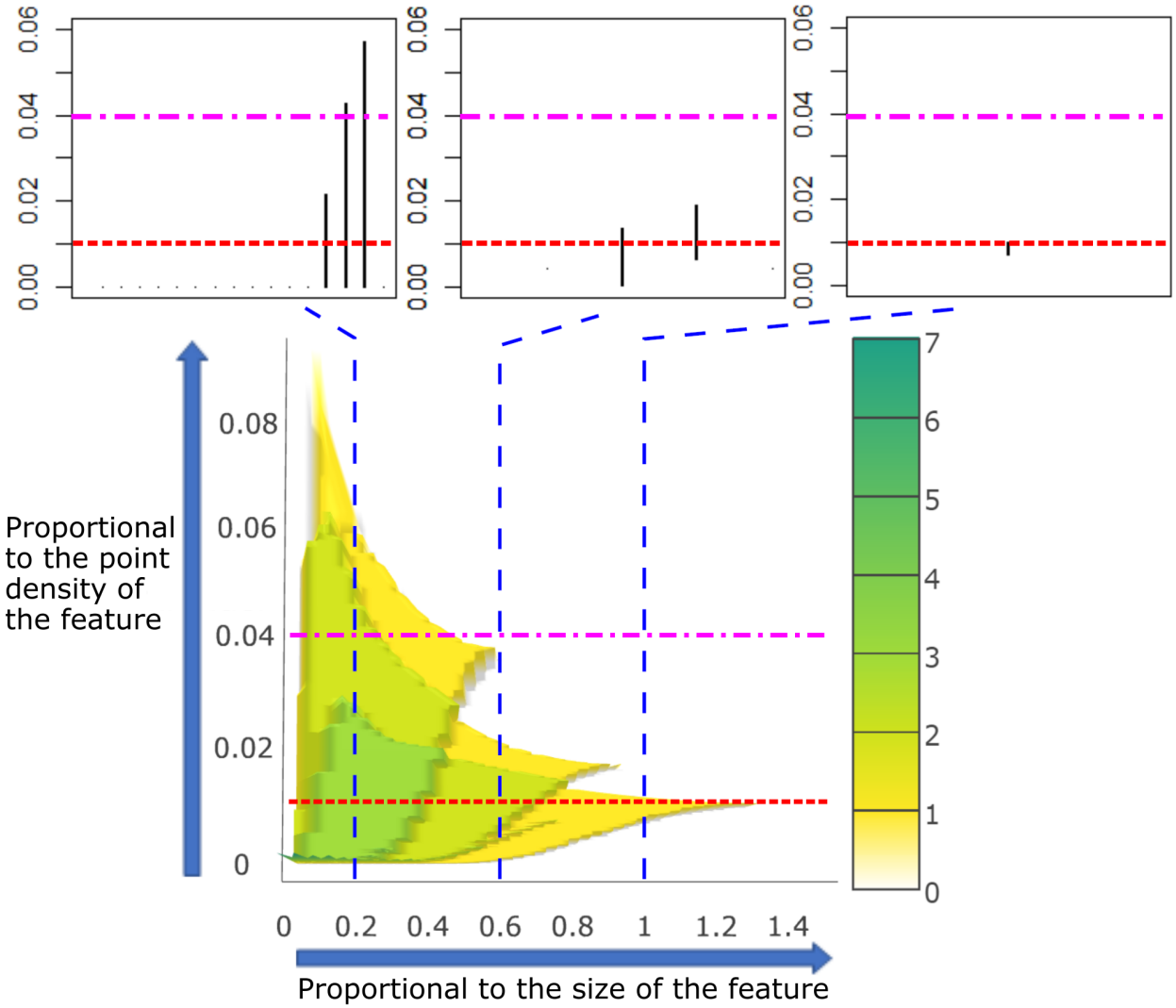
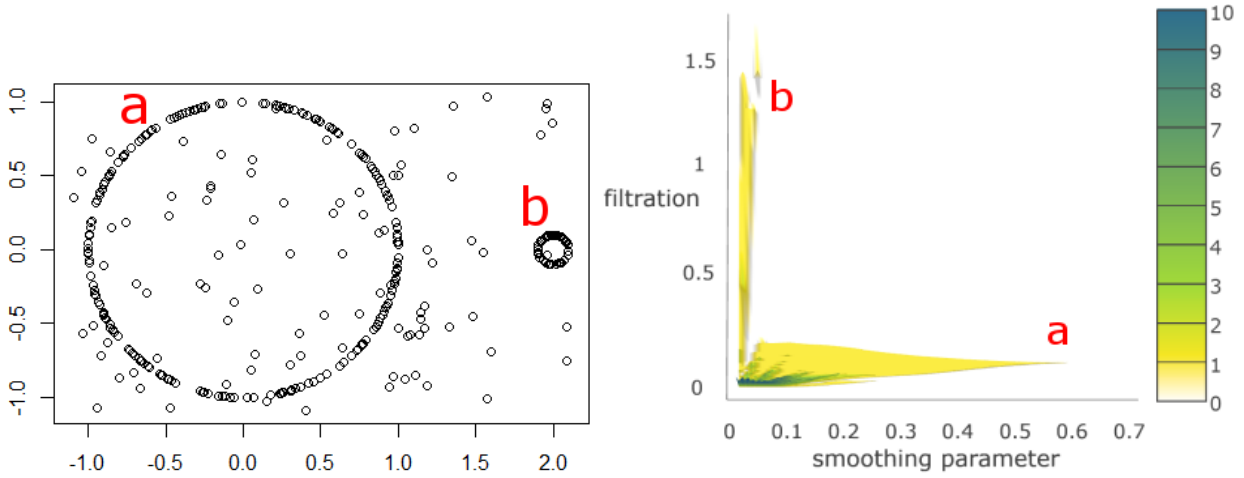


Figure 2.8: Barcode plots at smoothing parameters 0.2, 0.6 and 1, corresponding to vertical slices of the persistence terrace. Barcode plots traditionally use the x-axis for the filtration value so we rotated counterclockwise 90 degrees to match with the y-axis of the terrace.

2.3.1 Computational algorithm

We introduce the R package "pterrace" for creating persistence terraces and terrace area plots. The computation of a persistence terrace can be divided into three algorithmic steps:

$$\text{Point cloud data} \overset{Step\ 1}{\longrightarrow} \text{Barcodes} \overset{Step\ 2}{\longrightarrow} \text{Betti numbers} \overset{Step\ 3}{\longrightarrow} \text{Persistence terrace}$$



(a) Scatterplot (b) Persistence terrace (satellite view)

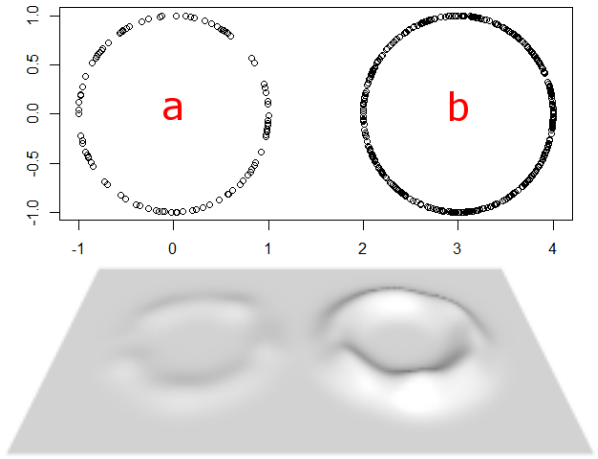
Figure 2.9: Scatterplot of the noise-added two circles data and β_1 persistence terrace. The two distinct terrace layers correctly suggest two significant topological features, which would be nearly impossible to detect using conventional Rips or Morse approaches.

First, we compute the barcodes for the Morse complexes corresponding to a pre-defined vector of smoothing parameter values, using a specified smoothing function. This is accomplished using a simple "for loop" and any of the pre-existing persistent homology software packages that includes Morse complex barcodes. We use the R package TDA (Fasy et al., 2014) for this step. Second (Algorithm 1 in Appendix), we fix a dimension k and for each fixed value of the smoothing parameter compute the Betti number β_k using the k-dimensional barcodes. Third (Algorithm2 in Appendix), we use the fact that for each fixed smoothing parameter value the function β_k just computed is a step function, only changing at the filtration values computed in the previous step, in order to assemble all the Betti numbers into the persistence terrace.

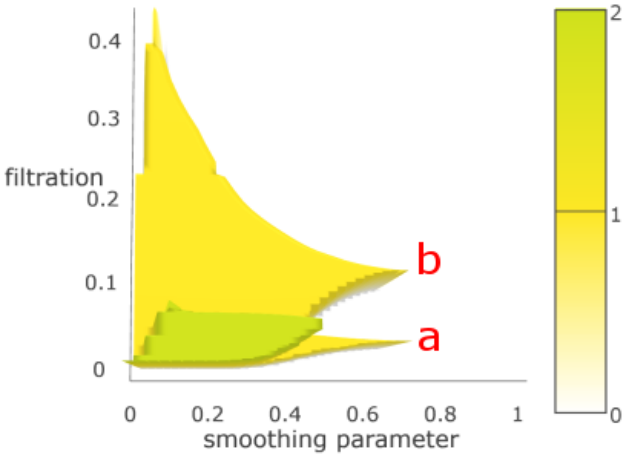
2.3.2 Detection of Features with Different Densities

We can use the persistence terrace to identify differences in the densities of data points that make up the various topological features in the point cloud. As discussed earlier, the

persistence terrace can often be visually decomposed into height-one layers that overlap with each other in various regions. The length of each height-one layer along the y-axis positively relates to the density of points in that feature. Indeed, high density means that the smoothing function will take large values, so the manifold it produces will be very tall over high density regions and consequently that portion of its level set topology will remain constant for a large range of filtration values.



(a) Scatterplot (top) and manifold from bandwidth 0.2 (bottom)



(b) Satellite view of β_1 persistence terrace

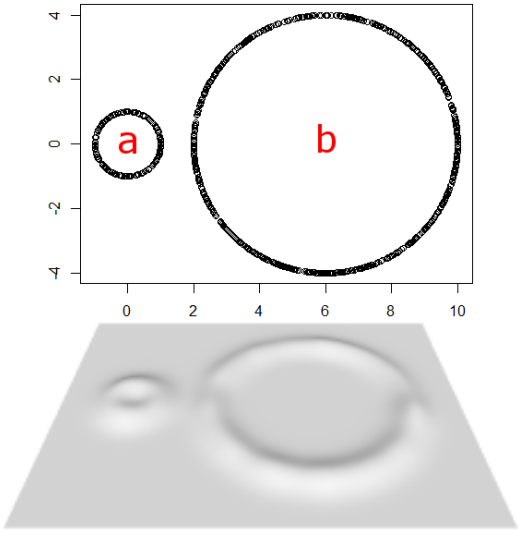
Figure 2.10: Scatterplot, manifold, and persistence terrace of two circles data with different densities. The high density circle b smooths to a high altitude volcano and appears as a persistence terrace layer stretching along the y-axis in the satellite view (layer b).

To illustrate this, we generate 100 and 400 random points from two equal sized circles. Figure 2.10a shows how the higher point density of the second circle is manifest as a higher altitude volcano-shaped manifold after smoothing with bandwidth 0.2. In the β_1 persistence terrace, we see two height-one terrace layers with a clear region of height-two overlap. Looking at the satellite view of the persistence terrace in Figure 2.10b, we see that both layers reach horizontally to equal maximal values of the smoothing parameter (which, as we discuss in the next subsection, stems from the fact that the circles have the same radius), but one layer reaches much further vertically along the filtration axis. This taller terrace layer represents the denser circle since for large values of the filtration parameter the super-level set of the

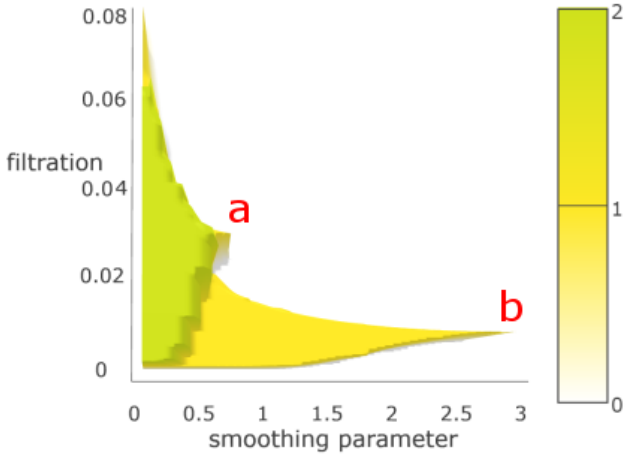
tall volcano will be an annulus (which, topologically, is a circle) while that of the short volcano will be empty. If we were to remove the dense circle from the point cloud and re-plot the persistence terrace, we would see this tall terrace removed (so the height-two overlap would drop down to height one) and the shorter terrace layer would remain unchanged. Thus the two topological signals have been completely disentangled and identified by their corresponding point densities.

2.3.3 Detection of Features with Different Sizes

We can also infer the size of topological features in a point cloud from the persistence terrace. Just as density is measured by how far a terrace layer stretches along the y-axis in the satellite view, size is measured by how far it stretches along the x-axis—that is, by the range of smoothing parameters for which the corresponding feature is detectable. Indeed, as the smoothing parameter increases, the corresponding manifolds become more rounded so the finer topological features of the super-level sets get washed away.



(a) Scatterplot (top) and manifold from bandwidth 0.5 (bottom)



(b) Satellite view of β_1 persistence terrace

Figure 2.11: Scatterplot, manifold, and persistence terrace of two circles with equal density but unequal radius data. The volcano manifold from the small circle a fills in more quickly as the smoothing parameter increases and appears as a persistence terrace layer stretching along a narrow stretch of the x-axis in the satellite view (layer a).

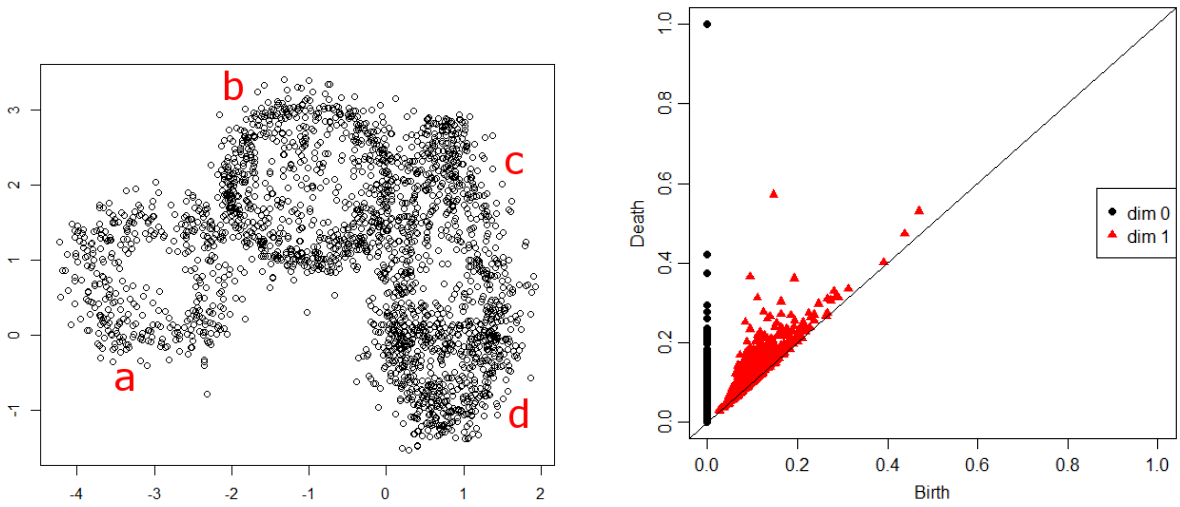
To illustrate this, we generate 200 points on a radius 1 circle and 800 points on a radius 4 circle so that the two circles have almost the same point densities. We see in Figure 2.11a that they smooth into volcanoes of similar height. The β_1 persistence terrace in Figure 2.11b clearly consists of two overlapping height-one layers. These layers reach to roughly comparable heights along the y-axis since the two densities agree, but one terrace layer reaches much further along the x-axis: the volcano coming from the smaller circle fills in more quickly than that of the larger circle as the smoothing parameter increases.

2.4 Simulation Study

The data sets in the previous section are deliberately very simple in order to focus the discussion and analysis of persistence terraces on specific attributes and behaviors. In this section we explore two data sets that are less artificial: one involves different shapes and more noise/distortions, the other comes from medical imaging.

2.4.1 Features with Noise

Here we demonstrate that the persistence terrace can be used to help analyze data that are both noisy and less uniform than in the previous section. We generate a planar point cloud with points clustering around four "holes" so that four topological loops should be present (see Figure 2.12a). Specifically, we place 400 points uniformly randomly on a 1.5×1.5 square then perturb these points with random $N(0, 0.15)$ noise (square a). We create 800 points around a radius 1 circle b: for the inside and outside of the circle, 400 points are generated that follow an exponential distribution with rates 4 and 10, respectively. We also create an equilateral triangle c and an isosceles triangle d, by placing 200 points randomly uniformly on each triangle edge, but then perturb these points with $N(0, 0.15)$ noise. The persistence diagram in Figure 2.12b is computed using the Rips complex. While some triangles in that diagram are far above the diagonal line, suggesting persistent loops in the data, there is far too much noise to be able to infer the correct number of loops, namely $\beta_1 = 4$.

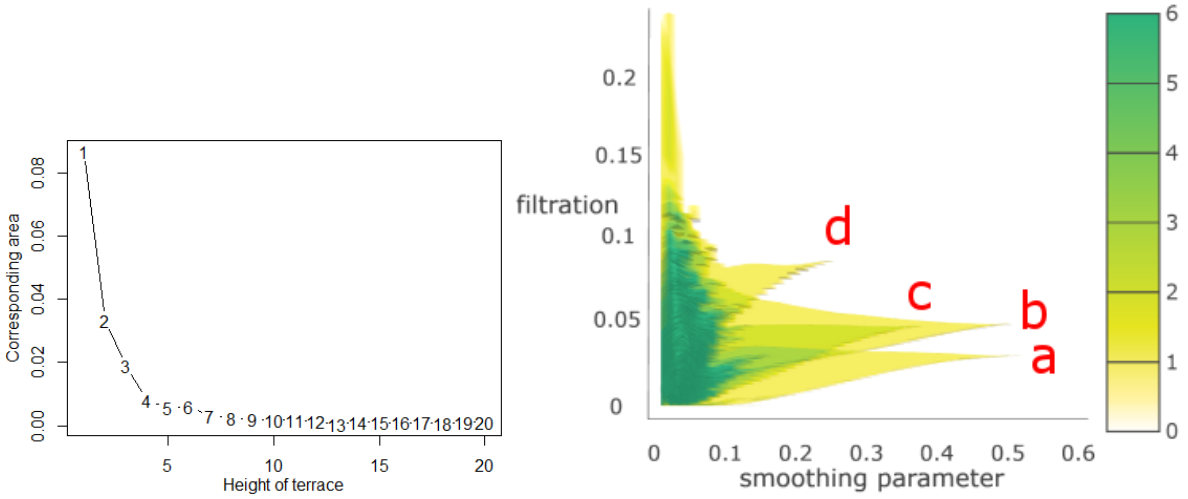


(a) Scatterplot (b) Persistence diagram

Figure 2.12: Scatterplot of four noisy shapes data and the Rips persistence diagram from which it is essentially impossible to infer precisely four significant topological features.

We compute the β_1 persistence terrace using 50 smoothing parameter values evenly distributed between 0.01 to 0.6. The persistence terrace detects up to 83 loops in the data, but most of these only occur for relatively small values of the smoothing parameter. We show the terrace area plot up to height 20 and persistence terrace in Figure 2.13. The rapid decline in the terrace area plot through height four suggests there are at least three significant features in the data. Although we expect the layers of height greater than three to be considered noise, we put all levels greater than six into a single category in the persistence terrace Figure 2.13b to avoid erroneous interpretation. The height-three layer in the persistence terrace is an overlap of the layers a, b, and c. The layer d is a distinct layer that is disjoint from the height-three region and should therefore be interpreted as another significant feature. Thus, by analyzing both the persistence terrace and the terrace area plot we find $4 = 3 + 1$ significant loops in the data, as expected. We note, however, that there is a genuine chance of misreading these plots—particularly as the data sets become more complicated—so these

plots should be viewed as tools to help study the topology of data, rather than a fool-proof methodology.



(a) Terrace area plot (b) β_1 persistence terrace

Figure 2.13: Terrace area plot and satellite view of β_1 persistence terrace for the noisy four shape data. There are four prominent layers, corresponding to the four shapes, though one of them (layer c) is stacked on top of another (layer b).

We can use the methods from the previous section to give a rough, qualitative description of the four loops identified by the persistence terrace. The two largest loops (layer a and b) have roughly the same size though one has slightly greater point density; this latter density matches the density of another, slightly smaller loop (layer c); finally, the fourth loop (layer d) is both smaller and denser than the rest. This description appears to accord with the square, circle, isosceles triangle and equilateral triangle, respectively.

2.4.2 Counting Muscle Fibers

Muscle tissue consists of tube-like shapes known as muscle fibers which are bundles of filaments ensheathed by a connective tissue known as endomysium. A cross-section of a muscle thus reveals a collection of semi-homogeneous regions (if one blurs the filaments together), one for each muscle fiber, that are delineated by walls made of endomysium. Counting the number of muscle fibers in a cross-sectional slice of a tissue sample can, therefore, be viewed as a topological problem: we need to compute the number of independent loops formed by

the endomysium. Since the sizes of the loops vary and the cross-sectional image is bound to have noise, this is a natural setting to apply the persistence terrace. For this example, we adapt Figure 1 of Mula et al. (2013), the muscle tissue cross-section image.

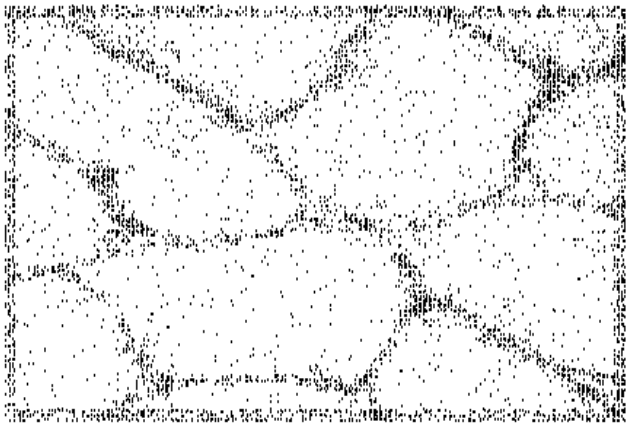


Figure 2.14: Scatterplot of 6,500 points sampled from a muscle tissue cross-sectional image, with added boundary lines to close the muscle fiber loops.

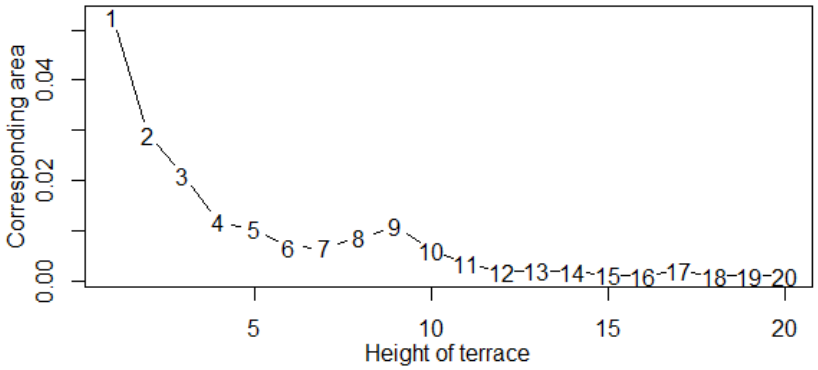
The first step in applying the persistence terrace method is to convert the cross-sectional image into a point cloud. Since we are interested in the loops made from the endomysium, we should sample points from the endomysium in the cross-sectional image. To do this, we first convert the original RGB image to a grayscale intensity image. Next, we randomly select 5,000 endomysial points from the grayscale image. We sample in proportion to the grayscale intensity so that darker pixels have a higher likelihood to be selected. Since the cross-sectional image contains several incomplete muscle fibers at the boundary that we would like to include in the count, we add lines around the boundary of the black-and-white image to artificially close off all the broken loops. For the particular cross-sectional image we are using here, this results in 11 muscle fibers, including these partial boundary fibers. We generate an additional 1,500 points by randomly sampling from the boundary. The resulting point cloud is shown in Figure 2.14.

Note that the muscle fiber cross-sections vary considerably in size, shape, and convexity. Note also that there are two types of noise: black pixels within the muscle fibers and white

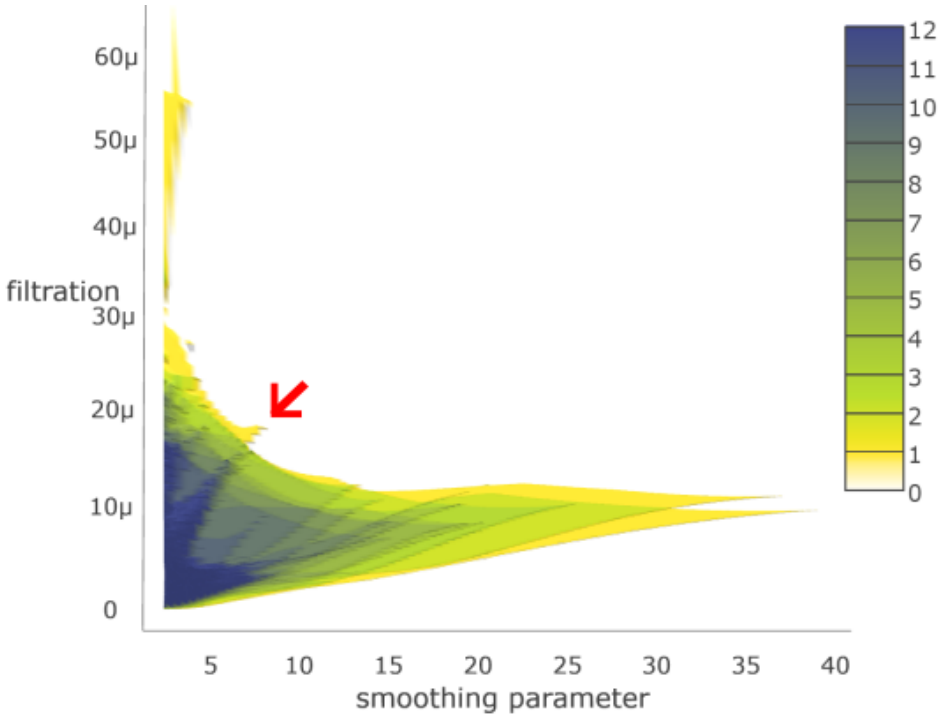
pixels within the endomysium, both due to sampling error. There is a further, more substantial complication to the data: while muscle fibers are packed together rather tightly, there are nonetheless small gaps where multiple fibers come together. Biologically, this means there are small chambers enclosed by endomysium that are devoid of filaments and thus not considered muscle fibers. From a data perspective, this means there are small white regions within the walls that lead to small loops in the point cloud that should not contribute to the muscle fiber count. The speckling noise renders Rips persistent homology inadequate while these gap loops make it nearly impossible to choose a single optimal smoothing parameter. These are both motivations for using the persistence terrace.

To build the β_1 persistence terrace, we use 100 smoothing parameter values between 2 and 40. Figure 2.15 shows the terrace area plot and persistence terrace. The terrace area plot shows that the minimum number of significant loops is either 9 or 10; the areas greater than 9 or 10 are small enough to be considered as noise. We color the heights greater than 12, considered to be noise, as a single category in the persistence terrace Figure 2.15b. Also, we can find a height-one triangular region, appearing around filtration value 20μ and smoothing parameter 8 (indicated by an arrow), which does not overlap with the height 9-10 terrace region. This suggests that we can count one additional small-sized high point density muscle fiber. Therefore, by considering the additional triangular region, we find 10-11 muscle fibers in total.

We can also see that there is a fairly prominent vertically oriented region extending to filtration value 60μ, giving terrace heights in the range 1-3. This implies that there is a small sized but high point density loop. The terrace region may correspond to the muscle fiber gaps (loops in the endomysium that do not enclose any filaments). It is difficult to get a precise reading of all the gaps in the scatterplot of Figure 2.14 because they are filled with noise pixels. These non-fiber loops are small in size but have comparable point density to the actual muscle fibers. Thus, they correspond to the top-left region in the persistence terrace.



(a) Terrace area plot



(b) β_1 persistence terrace

Figure 2.15: Terrace area plot and satellite view of the β_1 persistence terrace of the point cloud sampled from the muscle tissue cross-sectional image. The shape of the terrace area plot suggests at least 9–10 overlapping terrace layers, and the hand-drawn arrow in the persistence terrace reveals a non-overlapping layer, giving 10–11 total loops in the data.

While this portion of the analysis is more murky, it does seem that the persistence terrace is also able to detect and count some of the evident muscle fiber gaps.

2.5 Discussion

2.5.1 Algorithmic Improvements

To create a persistence terrace, we compute the Morse complex persistent homology for each value of the smoothing parameter specified in a pre-determined range. For large data sets, each persistent homology computation can be intensive. Fortunately, since these persistent homology computations are all independent, the procedure is *embarrassingly parallelizable*: just compute persistent homology for each smoothing value on separate processors then assemble. In our R package, we provide this parallel computing option.

We choose the range of smoothing parameter values heuristically then subdivide into equally sized intervals; if the terrace plot appears truncated horizontally, the range should be widened. A more subtle issue is choosing the number of intervals: too many intervals and the computational time becomes unreasonable, too few and the persistence terrace looks coarse and becomes difficult to read. Figure 1 in Appendix shows four persistence terraces computed from the Figure 2.5a data, using 25, 50, 75, and 100 smoothing parameter values. We see in that figure that the persistence terrace stabilizes with respect to this increase in resolution rather quickly. For more noisy and complicated data this stabilization will naturally require a greater number of smoothing parameter intervals.

2.5.2 Varying the Dimensions

In this chapter we have focused on the β_1 persistence terraces for planar point clouds. Extending from point clouds in $\mathbb{R}^2$ to arbitrary $\mathbb{R}^n$ is trivial: all steps in the algorithms already allow for this possibility, as does the qualitative analysis of the resulting persistence terraces. One can also extend from β_1 topological features (i.e., loops) to any other dimension. For higher dimensions, the overall analysis should be quite similar. Most applications of TDA in the literature focus on β_1 and β_0, so we did not explore β_2 or higher persistence terraces in our analysis but one could certainly find applications. On the other hand, we

try some experiments with β_0 persistence terraces and were unable to draw any conclusions. Since β_0 is the number of connected components, the β_0 persistence terraces should encode multi-scale clustering information about the point cloud, but it is limited by the fact that clustering involves more than just point density. We will explore this point in future work.

Chapter 3

Interface between Topological Data Analysis and Statistical/Machine Learning

3.1 Background

TDA provides information about the shape of data, which can reveal different aspects compared to existing approaches. Therefore, statistical/machine learning (SML) results may be improved by combining information obtained by TDA. However, persistent homology computation results have complex structures; they are given in intervals, and the number of intervals generated varies from dataset to dataset. These facts make it difficult to implement persistent homology results directly into SML methods because most of the latter take a vector as the input variable. There are two main ways of combining SML algorithms with persistent homology. First, a *kernel* method defines distances between persistence diagrams or embeds persistence diagrams in reproducing kernel Hilbert spaces (Kwitt et al., 2015; Kusano et al., 2016; Robins and Turner, 2016). Second, a *feature* method transforms the output of persistent homology into the standard vectorized format so that a real vector is associated with each barcode (Adcock et al., 2016; Kališnik, 2018). The computed output vectors can be used as input variables in SML. In this chapter, we suggest an improved approach for a feature method. We first present a list of features and discuss methods to choose a subset of features.

In the following subsections, we introduce the potential features using the following notation: consider a k-dimensional barcode with n number of bars, denote the left endpoint of the i^{th} bar by x_i, the right endpoint by y_i, the length of bar by $y_i - x_i$ and let y_{max} denote the right-most endpoint of any bar appearing in the barcode.

3.2 Features for Interface between TDA and SML

Adcock et al. (2016) suggest using the polynomial functions and Kališnik (2018) proposes using tropical coordinates on the barcode space. Tropical coordinates are defined based on tropical (max-plus) algebra, which studies the tropical semiring $(\mathbb{R} \cup \{\infty\}, \oplus, \otimes)$, with the operations $x \oplus y = \min\{x, y\}$ and $x \otimes y = x + y$ (Speyer and Sturmfels, 2009; Maclagan and Sturmfels, 2015). Carlsson and Verovšek (2016) define *min-plus and max-plus polynomials* as a linear combination of products of elements in the tropical semiring and *rational tropical functions* as a quotient of min/max-plus polynomial expressions, and Kališnik (2018) finds sets of the polynomials and rational functions that can be used for barcodes. For example, one of the *max-plus polynomials* used in Kališnik (2018) is $\max_{p<q<r}((y_p - x_p) + (y_q - x_q) + (y_r - x_r))$, where $r \leq n$. These approaches offer barcode features given in a real vector that can be used in the existing SML algorithms.

$$\text{Data} \rightarrow \text{Nested Complexes} \rightarrow \text{Set of Intervals} \rightarrow \text{Euclidean Vector}$$

In their examples, these authors show high classification rates using a small number of features. However, the suggested polynomial features in Adcock et al. (2016) are case-specific. The tropical coordinates suggested in Kališnik (2018) satisfy a stability property with respect to Wasserstein and bottleneck distances, but the size of an all possible set of features is large. A disadvantage of both of these approaches is that, for a given application, it is not known how to efficiently select the specific features to be used in the analysis. In addition, the topological characteristics of the features are not given. These make it difficult to apply and interpret the SML result. Instead, we suggest a new approach using a larger set of barcode features along with brief explanations.

3.2.1 Summary Statistics and Polynomials

The simplest way to summarize numeric values is using summary statistics: mean, standard deviation, quartile, maximum, minimum, etc. Most of these statistics are for univariate

variables, which cannot be directly applied to interval data. Intervals obtained from persistent homology cannot be analyzed by symbolic data analysis Billard and Diday (2007); the barcodes are not interval-valued symbolic random variables and the number of barcodes generated varies from dataset to dataset. We convert interval values into univariate variables and describe distributional information such as a center and spread. Examples of functions and univariate variables are listed in Table 3.1. Polynomials suggested in Adcock et al. (2016) are a subset of the combination of the listed functions and variables. Although the converted univariate variables may not contain entire information as barcodes, we expect the statistics to reveal different aspects. Researchers may investigate different aspects of intervals by combining functions and variables such as assigning different weights to the length by selecting different degrees and using functions such as log and exponential. As an example, let us assume that interval values vary from 0 to 1. If the degrees are set to be $a > b > 1$ for the variable $x_i^a(y_i - x_i)^b$, then the variable will give more weight to the shorter-length intervals. Also, other than simple statistics, we can also consider creating functions combined with certain conditions. For example, we can compute the sum of the k largest/smallest values instead of the sum of all values. Combination of functions such as the sum of $y_i - x_i$ where $x_i > Q_3(x_i)$ might uncover different information.

Table 3.1: List of possible functions and variables

Functions	Mean Quartile Standard deviation Order statistics (i.e. max/min) Sum Conditional sum
Variables	x_i y_i $y_i - x_i$ $y_{max} - y_i$ $x_{max} - x_i$ Combination of above variables (i.e. $x_i^a(y_i - x_i)^b$)

The basic statistics could also be computed for the persistence landscape. For example, one can count the number of peaks, or compute the area under the persistence landscape, or compute the same statistics as in Table 3.1 for the bars that are used to construct each persistence landscape.

3.2.2 Regression Coefficients

An alternative approach to bridging the gap between TDA and machine learning is comparing the compactness of data using dimension 0 barcodes obtained from the direct estimation method. For direct estimation, dimension 0 barcodes illustrate the life of components that are created at $\epsilon = 0$ and merged at $\epsilon = y_i$ (endpoints). Because all data are considered as separate components at the filtration value $\epsilon = 0$, the number of dimension 0 barcodes created is same as the number of data points. As the filtration value increases, the size of balls increases, and data points start to be connected. The endpoint of the dimension 0 barcode represents the filtration value at which the separate components are all merged into one connected component. Even though dimension 0 barcodes are given in interval form, it is sufficient to analyze only endpoint values because they all start at the same place. Thus, for the regular TDA's dimension 0 analysis, we can convert the interval data of (x_i, y_i) into the single value data y_i.

Once the number of the connected components is computed, we can plot them according to the filtration. Such a plot would visualize the compactness of the data. For example, if we analyze two datasets with different densities, then the number of connected components will decrease faster for the dense dataset as the filtration increases. Moreover, we can fit a model to the number of connected components. The coefficients of the fitted model represent the density or the compactness of data. We can build classifiers with coefficients values using existing classification methods. The endpoint curve of dimension 0 ($x_i = 0$) is used to compare compactness of brain networks in Lee et al. (2011).

We show how regression coefficients can be applied to compare compactness using a simulation study. We apply regular TDA to data with four different densities. For each setting, we place 360 data points on the three-by-three square. As shown in Figure 3.1, different numbers of random bivariate uniform data are assigned to each one-by-one square. The data points of setting 1 are concentrated on the left-upper part of the square, and the points of setting 4 are uniformly distributed across the region. The data points get less compact as the setting numbers increase.

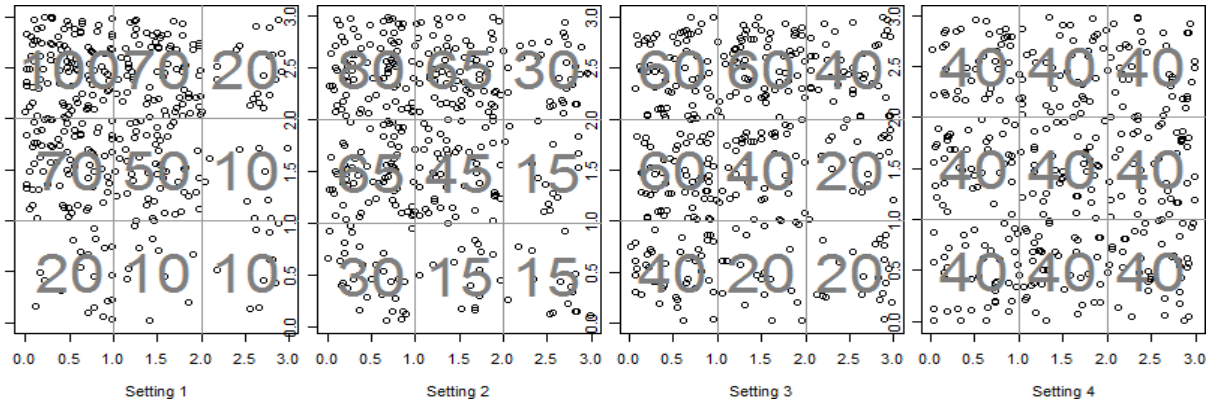


Figure 3.1: Scatterplots of four density settings

We generate 20 sets of data for each setting and run regular TDA with dimension 0 barcodes as in the previous subsection. Figure 3.2 shows the number of connected components according to the filtration value. Under setting 1 (red), data are connected quickly until around 30 connected components are left, and the merging speed gets slower after that. This agrees with the distribution of setting 1 that has three one-by-one squares of 10 points in the lower-right corner. On the other hand, setting 4 trends (purple) decrease at a constant rate and look like straight lines. This is because the 360 data points are uniformly distributed on the three-by-three square.

We fit four models; first-, second-, and third-degree polynomial and logit models. Figure 3.3 shows the coefficients of the fitted models. Compactness of data is summarized into numeric coefficient values. The first-degree polynomial and logit models successfully separate data with different densities just with two coefficients.

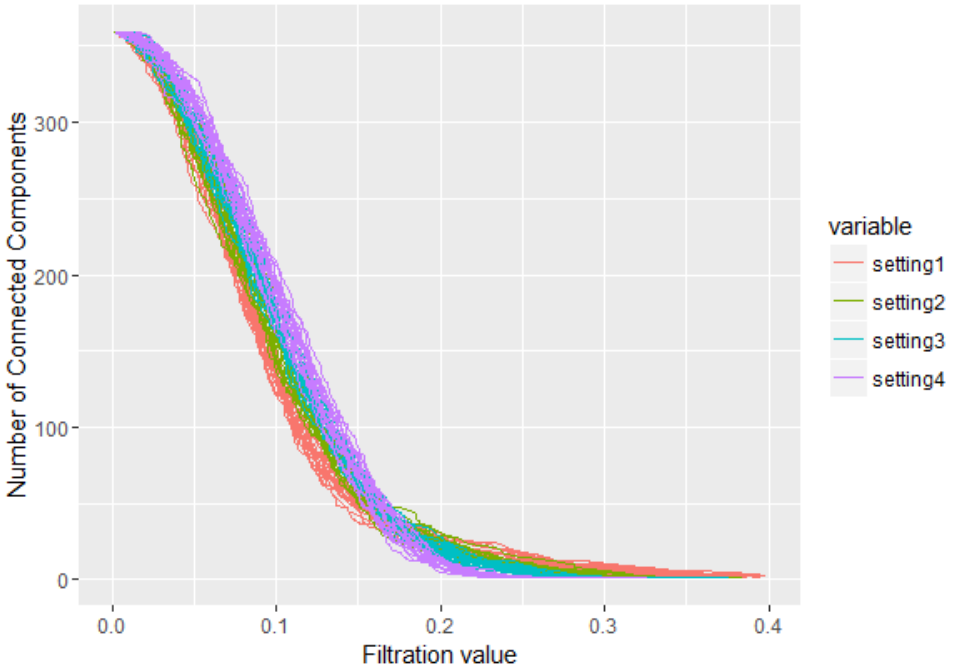
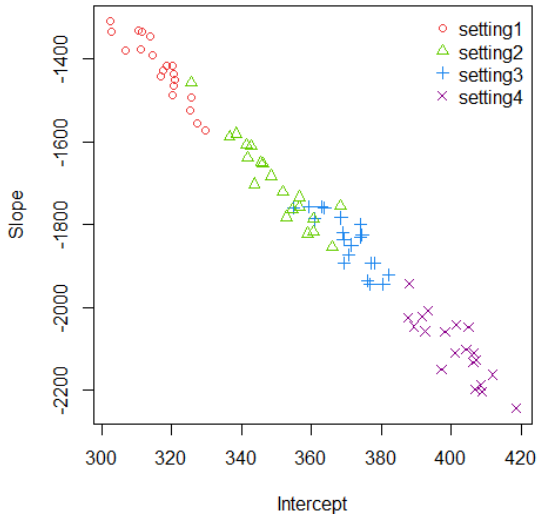


Figure 3.2: Number of connected components of density data

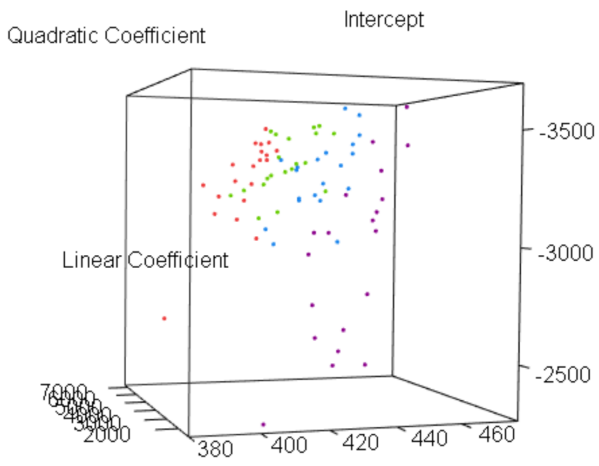
3.3 Feature Selection and Learning Methods

The number of features that can be generated in an example such as that of Section 3.2 can easily exceed the number of observations. Feature selection methods should be used to select a subset of features for constructing (prediction/classification/clustering) model. In many applications, it has become necessary to use feature selection methods. These methods have been used to 1) prevent over-fitting of the model, 2) improve model performance, 3) lower computational cost and 4) gain better insight from data (Guyon and Elisseeff, 2003; Fan and Fan, 2008; Li et al., 2017). Although the art/science of choosing barcodes themselves is still in the early stages, there exist several approaches that can be utilized for barcode feature selection.

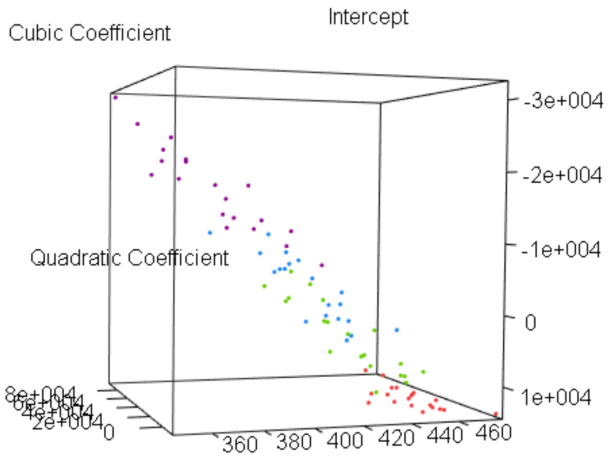
Feature selection methods generally focus on reducing two different types of features; *redundant* and *irrelevant* variables (Guyon and Elisseeff, 2003). If features contain similar



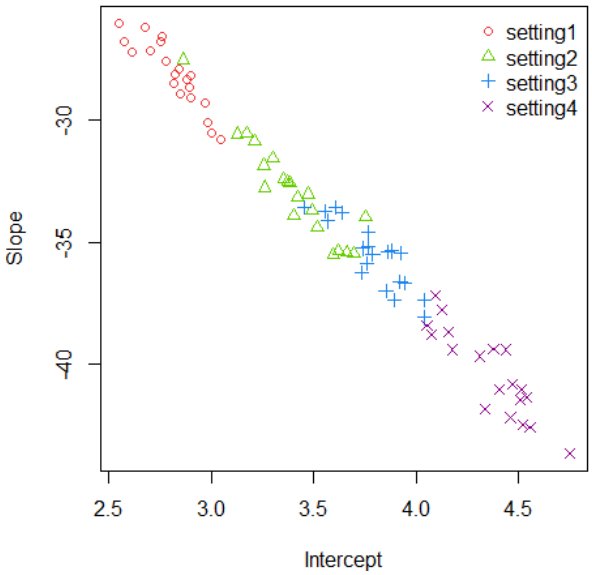
(a) First degree polynomial



(b) Second degree polynomial



(c) Third degree polynomial



(d) Logit model

Figure 3.3: Coefficients of fitted models

information about the data, then they are redundant. In many cases, performance of methods improves when redundant features are removed. One of the ways to identify redundant features is through the use of correlation (Hall, 1999). For example, when two features are highly correlated, then one of the features that is more highly correlated with the other variables can

be dropped. Backwards elimination can also be applied to remove irrelevant features (Miller, 1990); this is a greedy algorithm that removes features one at a time, choosing a single feature at each step whose removal maximizes the accuracy rate. As features are removed in this manner, accuracy rates generally first increase (as over-fitting is reduced) and then decrease (as under-fitting begins to take over). The peak of this backwards elimination curve indicates a reasonable balance and identifies a moderately sized collection of features that work well together for classification.

3.4 Application to Fingerprint Classification

Near the end of the 19th century, Sir Francis Galton introduced a systematic framework for fingerprint analysis (Galton, 1892). One component of his work was to divide all fingerprints into three classes: arch, loop, and whorl. This classification, often with refinements (such as subdividing arches into plain and tented types, and dividing whorls into singles and doubles) is still used by nearly every fingerprint classification scheme today. Automated fingerprint classification is useful for fingerprint matching algorithms, since it reduces the search space involved; it also provides a fertile testing ground for more general explorations in pattern analysis. There is a wide variety of approaches to automated classification, tested on various real and simulated fingerprint databases; see the survey articles Yager and Amin (2004); Ahmad and Mohamed (2009) and the book chapter of Maltoni et al. (2009) for more background and context, an overview of proposed methodologies, and a comparison of performances.

3.4.1 Fingerprint Data

We test our methods on the National Institute of Standards and Technology Special Database 27 (NIST SD-27). The database is a forensically-oriented fingerprint database designed to help researchers develop and hone matching algorithms for fingerprints of varying quality. It

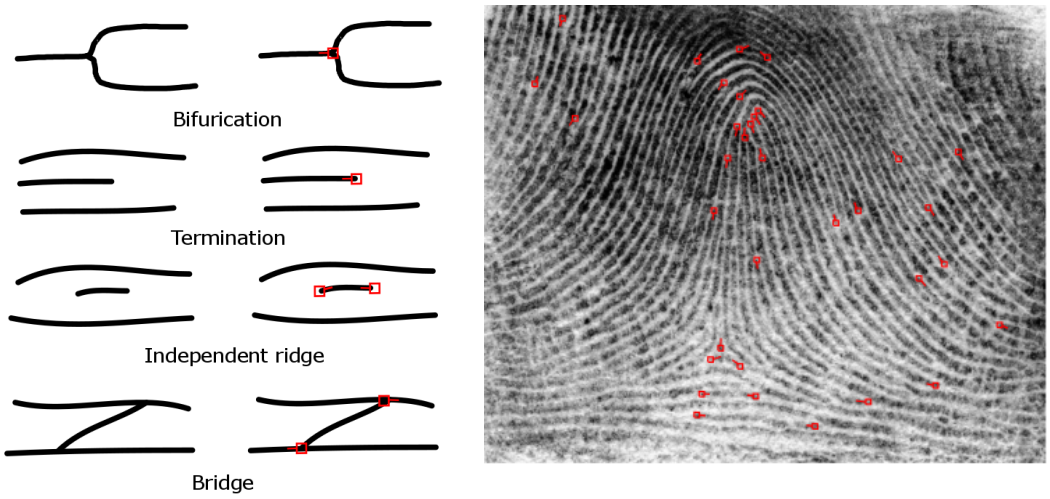
was originally released in 2000 and then re-released in 2010 with higher resolution images. The database is composed of 258 fingerprint entries, each containing the following data:

- A *latent* fingerprint image obtained from a crime scene. The finger could be any of the ten digits, and the quality varies from "good" to "bad" to "ugly".
- A police department ink-roll, called a *tenprint*, of the same finger on the same individual as the latent print. While these images are higher quality than even the best latents, they still contain noise, noticeable imperfections, and cropping artifacts inherent to the ink-roll process that pre-dates modern digital fingerprint imaging technology.
- A fingerprint class identified by a human expert: plain arch, tented arch, right slant loop, left slant loop, whorl, or unclassifiable (see Figure 3.4 for some examples).
- Four sets of minutiae points that were hand-identified by experts: (1) all the minutiae that were directly discerned on the latent print (called *ideal latent minutiae*), (2) all the minutiae that were directly discerned on the tenprint (called *ideal tenprint minutiae*), (3) the latent minutiae that were identified with corresponding tenprint minutiae (called *matched latent minutiae*), and (4) the tenprint minutiae that were identified with corresponding latent minutiae (called *matched tenprint minutiae*). Each minutiae point is recorded by its coordinates in the fingerprint image; also recorded is the *orientation*, a radial measure of the direction of the bifurcation/termination where the minutiae point occurs (see Figure 3.5).

Some fingerprint entries are missing an ink-roll image or have the class listed as unclassifiable; we remove these from the database, resulting in 245 of the original 258 entries. In some cases the fingerprint expert could not decide on a single class and so listed multiple possible classes; in such cases we use the first listed class as that is the one in which the expert had the greatest confidence. The distribution of classes in the 245 fingerprints is then: 5.3% arches, 58.4% loops, and 36.3% whorls. According to Wilson et al. (1993), the naturally occurring probabilities of these classes in a general human population are 6.6% arches, 65.5%

(a) Arch (b) Loop (c) Whorl

Figure 3.4: Examples of the three fingerprint classes from the dataset NIST SD-27



(a) Types of minutiae (b) Minutiae on fingerprint image

Figure 3.5: Minutiae

loops, and 27.9% whorls, so this NIST database is fairly representative in this regard. The number of minutiae points in these 245 fingerprints ranges from 48 to 193.

As can be seen in Figure 3.4, the ink-roll images in this database include varying amounts of white space and regions of the finger (often extending past the first joint) and they

frequently contain extraneous markings. The only image-processing we perform is manual cropping to mostly eliminate the white space and focus the image on the region of the fingerprint above the main horizontal crease in each finger. Some examples of cropped images are shown in Figure 3.6.

3.4.2 Persistent Homology Modeling

Minutiae-based Approach

A natural TDA approach is to view fingerprint minutiae points as a point cloud in $\mathbb{R}^2$—that is, to use minutiae point locations as the input for persistent homology. As we later discuss in Section 3.4.4, this yields a rather mediocre classification performance. A remarkable improvement is obtained by incorporating the minutiae point orientations. The key insight for how to do this is that persistent homology allows the point cloud to live in any metric space, not just Euclidean space $\mathbb{R}^m$. Minutiae point orientations are simply angles, so they are naturally viewed as points on the unit circle S^1. The minutiae points on a given fingerprint then form a point cloud in the manifold $\mathbb{R}^2 \times S^1$, which is a higher-dimensional analogue of the cylinder. There are various natural choices for endowing this product space with a metric. The choices we use are based on the ℓ^1 metric (*taxicab* or *Manhattan*), the ℓ^2 metric (*Euclidean*), and the ℓ^3 metric. Given the set of N minutiae points

$$\mathbf{p_1} = (a_1, b_1, \theta_1), \ \ldots, \ \mathbf{p_N} = (a_N, b_N, \theta_N) \in \mathbb{R}^2 \times S^1$$

of a fingerprint, we first normalize by replacing each a_i with

$$\frac{a_i - \min_{1 \le j \le N}\{a_j\}}{\max_{1 \le j \le N}\{a_j\} - \min_{1 \le j \le N}\{a_j\}}$$

and similarly for b_i, and each θ_i with $\frac{\theta_i}{\max_{1 \le j \le N}\{\theta_j\}}$, so that all coordinates and angles are between 0 and 1. We then define five different metrics computing distances between any pair of these normalized points:

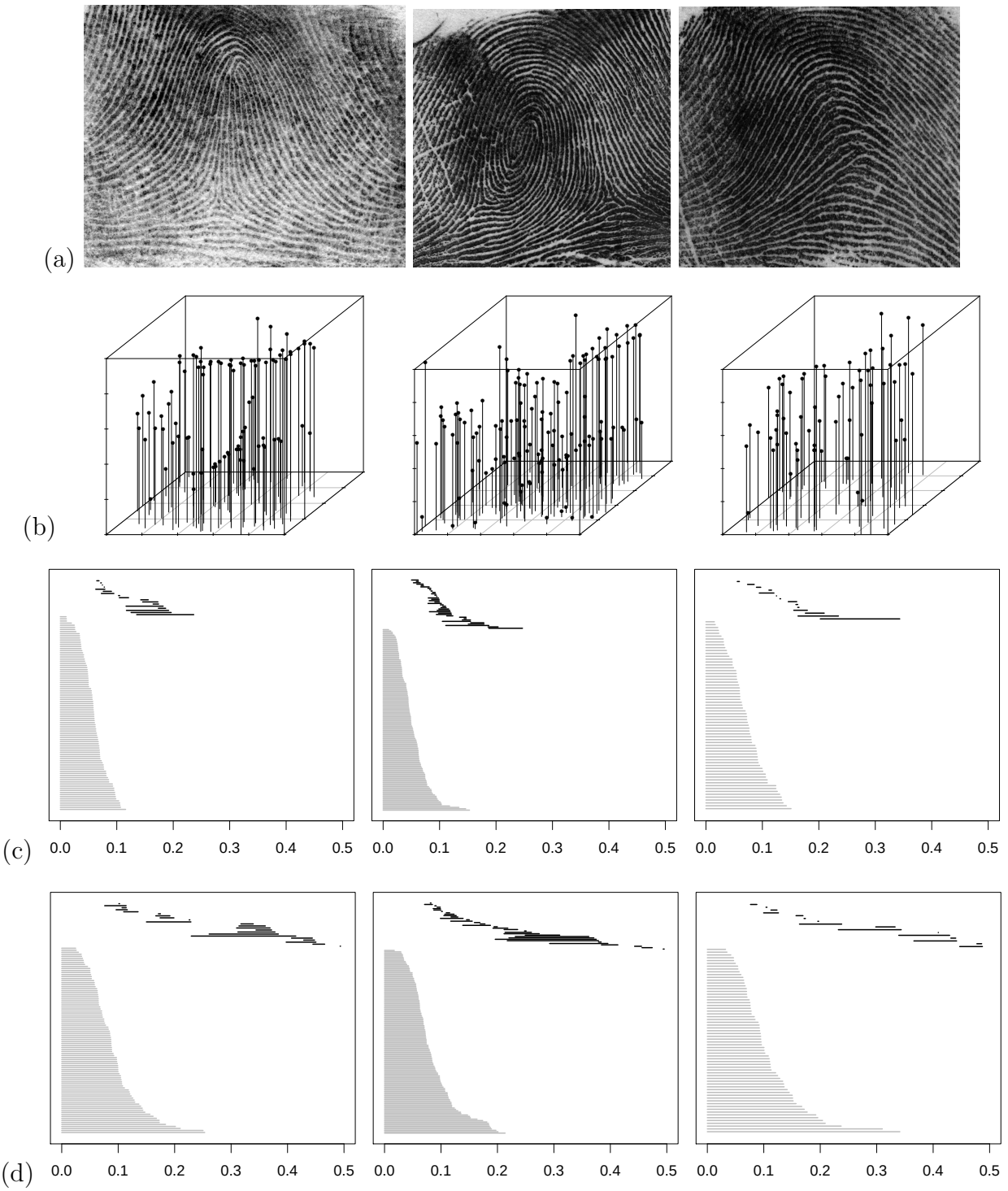


Figure 3.6: (a) Cropped images of a loop (left), whorl (middle), and arch (right). (b) Scatterplots of the corresponding normalized minutiae coordinates (the vertical axis is the orientation, so the top and bottom squares should be identified). (c) The 0-dimensional (gray) and 1-dimensional (black) barcodes for the unoriented minutiae point clouds in $\mathbb{R}^2$. (d) The barcodes for the minutiae point clouds in $\mathbb{R}^2 \times S^1$ with the metric d_2 defined earlier. Precisely interpreting these barcodes is not necessary; for the purposes of supervised learning we simply need that the barcodes reflect *some* relevant global geometric structure in the fingerprints.

$$
\begin{aligned}
d_1(\mathbf{p_i}, \mathbf{p_j}) &= |a_i - a_j| + |b_i - b_j| + \theta_{ij} \\
d_{1,\frac{1}{3}}(\mathbf{p_i}, \mathbf{p_j}) &= \frac{1}{3}(|a_i - a_j| + |b_i - b_j|) + \frac{2}{3}\theta_{ij} \\
d_{1,\frac{2}{3}}(\mathbf{p_i}, \mathbf{p_j}) &= \frac{2}{3}(|a_i - a_j| + |b_i - b_j|) + \frac{1}{3}\theta_{ij} \\
d_2(\mathbf{p_i}, \mathbf{p_j}) &= \sqrt{(a_i - a_j)^2 + (b_i - b_j)^2 + \theta_{ij}^2} \\
d_3(\mathbf{p_i}, \mathbf{p_j}) &= \sqrt[3]{(a_i - a_j)^3 + (b_i - b_j)^3 + \theta_{ij}^3}
\end{aligned}
$$

where

$$
\theta_{ij} = \begin{cases} |\theta_i - \theta_j| \text{ if } |\theta_i - \theta_j| \leq \frac{1}{2} \\ 1 - |\theta_i - \theta_j| \text{ if } |\theta_i - \theta_j| > \frac{1}{2} \end{cases} .
$$

For each of the 245 fingerprints of interest in NIST SD-27, we compute 12 distinct barcodes: the 0- and 1-dimensional persistent homology for the minutiae point clouds in $\mathbb{R}^2$ (with Euclidean metric and orientations ignored) and in $\mathbb{R}^2 \times S^1$ using each of the five different metrics listed above. We use the R package TDA (Fasy et al., 2014) for persistent homology computation. See Figure 3.6 for an example of persistent homology compuation results.

Image-based Approach

The point cloud approach to persistent homology is just one of several options. Another form of persistent homology uses a discrete variant of Morse theory. We apply this Morse-based method to fingerprint data as follows:

1. read in the 245 cropped grayscale JPEG fingerprint ink-rolls in NIST SD-27, invert them (so that the background is black instead of white) and store as real-valued matrices;

2. normalize each matrix by first subtracting off the minimal matrix value and then dividing the new entries by the new maximal value;

3. compute the 0- and 1-dimensional superlevel set persistent homology barcodes of the surface defined by each normalized matrix (see Figure 3.7, top row), where the grid resolution is provided by the matrix itself (i.e., each grid square is a single matrix entry, which corresponds to a single JPEG pixel).

In principle these barcodes record the global topology of the fingerprint ridge pattern, but there is so much noise and so many minor fluctuations that it is nearly impossible to see this. Regardless, the supervised learning classification pipeline works better when accessing finer geometric information, which we achieve by a novel "slanting" method that we introduce here. After normalizing the JPEG image matrices in steps 1 and 2 above, we obtain additional barcodes as follows:

4. multiply the normalized matrix by the linear function $f(x, y) = y$ then compute the 0- and 1-dimensional barcodes for the superlevel set persistent homology (see Figure 3.7, middle row); do the same for the function $f(x, y) = x$;

5. threshold the normalized matrix by setting all values above the mean value of the matrix to 1 and all values below the mean to 0, then multiply by the function $f(x, y) = y$ and compute the 0- and 1-dimensional sublevel set barcodes (see Figure 3.7, bottom row); do the same for the linear function $f(x, y) = x$ and the non-linear function $f(x, y) = xy$.

This yields 12 barcodes for each fingerprint—six 0-dimensional and six 1-dimensional—just as we have with the minutiae-based approach. The motivation for this slanting process is that the persistent homology of the sublevel/superlevel sets after slanting is related to the "sweep across" persistent homology of the ridge curves and so measures the *tortuosity* of the ridges, not just their global topology. The thresholding step accentuates the ridge pattern and provides another way to increase the number of barcodes available (as does alternating between superlevel sets versus sublevel sets).

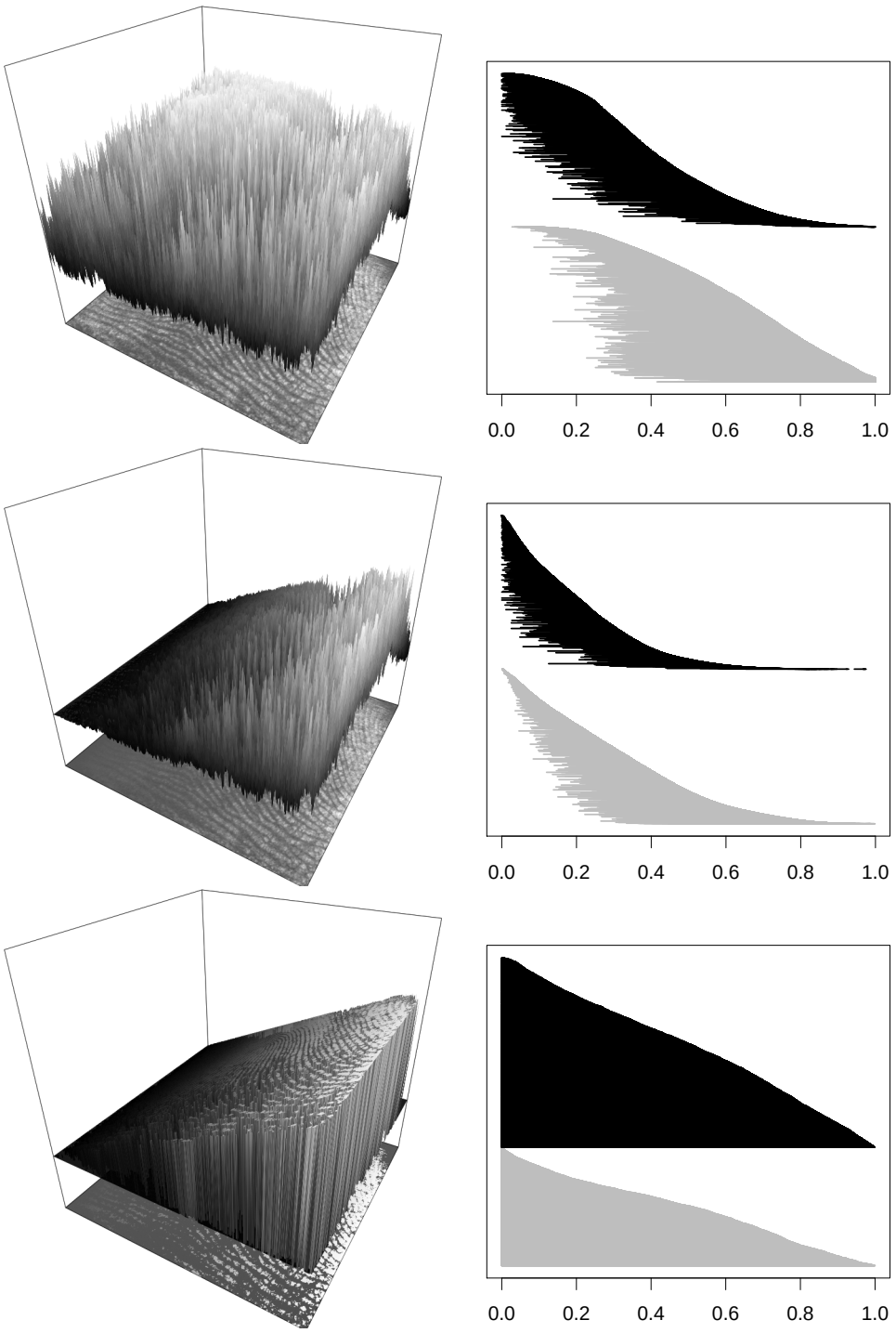


Figure 3.7: Top row: an ink-roll JPEG, after normalizing and inverting, viewed as a 3D surface, and the 0-dimensional (gray) and 1-dimensional (black) barcodes of the superlevel sets. Middle row: the same image after slanting by the function $f(x, y) = y$, and its superlevel set barcodes. Bottom row: same slanting, but first the image is thresholded to convert from grayscale to black-and-white, and here the barcodes use sublevel sets. We employ these variants (and the others described earlier) in an effort to access as much geometry of the ridge pattern as possible.

3.4.3 Features Used in Classification

We use the same notation as Section 3.2. We draw our feature vectors from the following collection of real-valued functions defined on the space of barcodes:

- *Statistical features*

  The distribution of bars provides the following statistical features:

  $$g_1 = \text{mean}\{x_i\} \quad g_2 = \text{mean}\{y_i\} \quad g_3 = \text{mean}\{y_{max} - y_i\} \quad g_4 = \text{mean}\{y_i - x_i\}$$
  $$g_5 = \text{median}\{x_i\} \quad g_6 = \text{median}\{y_i\} \quad g_7 = \text{median}\{y_{max} - y_i\} \quad g_8 = \text{median}\{y_i - x_i\}$$
  $$g_9 = \text{SD}\{x_i\} \quad g_{10} = \text{SD}\{y_i\} \quad g_{11} = \text{SD}\{y_{max} - y_i\} \quad g_{12} = \text{SD}\{y_i - x_i\}.$$

- *Polynomial features*

  We use the following polynomial features:

  $$f_1 = \sum_{i=1}^{n}(y_i - x_i) \qquad f_2 = \sum_{i=1}^{n} n(y_i - x_i)$$
  $$f_3 = \sum_{i=1}^{n}(y_{max} - y_i)(y_i - x_i) \qquad f_4 = \sum_{i=1}^{n} n(y_{max} - y_i)(y_i - x_i)$$
  $$f_5 = \sum_{i=1}^{n}(y_{max} - y_i)^2(y_i - x_i)^4 \qquad f_6 = \sum_{i=1}^{n} n(y_{max} - y_i)^2(y_i - x_i)^4.$$

  The even-numbered features (f_2, f_4, f_6) include the number of intervals n so that they can consider spread of variables. Also, features using higher degree terms (f_5 and f_6) give higher weight to the longer length bars compared to f_3 and f_4.

- *Regression coefficients*

  We sort the endpoints $y_1, \ldots, y_n$ in decreasing order and fit a degree ℓ polynomial regression model. The coefficients $c_0^\ell, \ldots, c_\ell^\ell$ are used as features. We use $\ell = 1$ and $\ell = 2$.

In total this yields $23 = 6 + 5 + 12$ features for each barcode. Since we have 12 minutiae-based barcodes and 12 image-based barcodes, we obtain $552 = 23(12 + 12)$ features for each fingerprint, though some features will be identically zero so we omit these (and many of these features are highly correlated—a point we return to shortly). As is common practice in machine learning, we normalize the feature vectors so that each has mean zero and standard deviation one.

We use linear discriminant analysis (LDA) classifiers (Fisher, 1936), one of the classic classifier, to show that the classification result is mostly due to the features, not a state-of-the-art classification algorithm itself. Since our database is rather small (there are 143 loops, 89 whorls, and only 13 arches), two important steps are necessary: (1) feature selection is first performed to mitigate the curse of dimensionality, and (2) rather than subdividing into training and testing subsets, we use the leave-one-out-cross-validation (LOOCV) method (Stone, 1974; Geisser, 1975; Picard and Cook, 1984). That is, after fixing an appropriate subset of the 552 features, we consider each fingerprint F_i and train an LDA classifier on the 244 complementary fingerprints $\{F_j\}_{j\neq i}$ then attempt to classify F_i; the LOOCV accuracy rate is the number of correct classifications, for $i = 1, \ldots, 245$, divided by 245. This is a standard machine learning technique when dealing with small data sets (James et al., 2013).

For feature selection, we employ two established techniques. First, once a collection of features of interest has been chosen manually (e.g., all the minutiae-based barcodes or all the image-based barcodes), we use the `findCorrelation` function in the R package "caret" to remove redundant features that are highly correlated with other features, based on a cutoff value 0.9. Next, we perform backwards elimination to remove irrelevant features. Although this tends not to find the optimal subset of features for classifying, it is a reasonable approximation given the computational infeasibility of searching all 2^{552} possible subsets.

3.4.4 Classification Results

Since the dominant fingerprint class in this database is the loop, with 143 out of 245 occurrences, the baseline accuracy rate that all approaches here should be compared to is 58.4%. The accuracy rates we obtain using persistent homology are summarized in Table 3.2, though first some explanation is in order. For each collection of features, we choose a cutoff value such that after removing the highly correlated features within this collection determined by `findCorrelation`, there are between 70 and 90 features remaining (except for "unoriented minutiae features," meaning the minutiae point clouds in $\mathbb{R}^2$, since there are only 46

features before removing the highly correlated ones). We then run backwards elimination on these latter features to thin them down further and select the subset of features with the highest accuracy rate among those tested during the backwards elimination. This is the "peak accuracy rate" reported in the table. The "number of features" indicates the size of the subset(s) found by backwards elimination that yields this peak accuracy rate (in some cases multiple subsets achieved the same peak accuracy rate). The minutiae-based features are mostly much stronger than the image-based features, except a few of the thresholded image xy-slant features are fairly competitive. Also, the unoriented minutiae-based features are not as strong as any of the oriented variants.

Table 3.2: The peak accuracy rates obtained when selecting various sets of features, removing highly correlated ones, then performing backwards elimination on the remaining ones. The rate listed is the maximum obtained this way for each group, and the number of features is the size of the subset(s) of features achieving this rate.

	Peak accuracy rate	Number of features
All 552 features	93.1%	32
The 276 0-dimensional features	82.0%	19, 25, 28
The 276 1-dimensional features	93.1%	32, 33
The 276 minutiae-based features	91.4%	48
The 276 image-based features	77.1%	37, 40
The 46 unoriented minutiae features	62.9%	11

It is difficult to pin down an accuracy rate for state-of-the-art methods appearing in the literature, since different data sets are used, different preprocessing steps are permitted, different numbers of classes are considered, etc. Useful tables of accuracy comparisons among a wide range of methods are given in Maltoni et al. (2009) and in Yager and Amin (2004). Most of these methods use larger databases, which would improve a supervised learning method such as ours, but they also allow four or five classes instead of our three, which certainly makes the classification problem harder. Regardless, we get a coarse estimate by noting that all these published accuracy rates range between 81% and 97%. It should be noted, however, that some of these reported scores in other literatures are slightly inflated

Table 3.3: The confusion matrices for the two best classifiers using 32 features selected from 552 features (top) and 1-dimensional features (bottom).

		Predicted			
		Loop	Arch	Whorl	Total
Actual	Loop	138	2	3	143
	Arch	6	7	0	13
	Whorl	6	0	83	89
	Total	150	9	86	245

		Predicted			
		Loop	Arch	Whorl	Total
Actual	Loop	137	0	6	143
	Arch	5	8	0	13
	Whorl	6	0	83	89
	Total	148	8	89	245

compared to what we report in Table 3.2. For some fingerprints, it is difficult to classify them into a single class. In these cases, fingerprints are given multiple classes. Some studies considered an observation to be correctly classified if the algorithm yields any of the listed classes fingerprints labelled with multiple classes. On the other hand, we only accept the first class listed by the NIST experts. Also, some studies allow a certain rejection rate, meaning a fixed percentage of difficult fingerprints are removed from the database prior to computing an overall accuracy rate—we use a 0% rejection rate: every fingerprint in NIST SD-27 that has a class indicated and a matching JPEG image is included.

The confusion matrices in Table 3.3 show that for our two best classifiers, nearly half of the arch fingerprints are misclassified as loops but most loops and whorls are correctly classified. A larger training set could help with this issue, but inspecting fingerprints (B) and (C) in Figure 3.4 shows that loops and arches can in fact appear quite similar.

We see from Table 3.4 that for the best subset of features we found among the 552, most but not all are 1-dimensional, there is a mix of minutiae-based and image-based, and certain functions (such as $g_{11} = \text{SD}\{y_{max} - y_i\}$) show up much more frequently than others. The

Table 3.4: With the notation for our features introduced in Section 3.4.3, these are the 32 features selected from among all 552 that achieve our best rate, 93.1%.

0-dimensional							
Minutiae cloud d_3 metric	f_5	g_{11}					
Image surface	g_{11}						
Image x-slant	f_5						
Image thresholded xy-slant	g_{11}						
1-dimensional							
Minutiae cloud d_1 metric	g_8						
Minutiae cloud $d_{1,\frac{1}{3}}$ metric	f_5	g_8					
Minutiae cloud $d_{1,\frac{2}{3}}$ metric	f_3	f_6					
Minutiae cloud d_2 metric	c_1^1	g_1	g_3	g_4	g_8	g_{10}	g_{12}
Minutiae cloud d_3 metric	c_2^2	f_6	g_3				
Image surface	f_5						
Image y-slant	g_3						
Image thresholded x-slant	f_2	g_{11}					
Image thresholded y-slant	c_1^2	c_2^2	g_4	g_{11}	g_{12}		
Image thresholded xy-slant	c_2^2	g_1	g_2				

strongest features tend to be 1-dimensional, though the 0-dimensional features provide some crucial arch versus loop/whorl separation.

3.5 Discussion

3.5.1 Stability of Suggested Features

Among the features listed in Section 3.2, only the tropical coordinates of Kališnik (2018) have been proven to satisfy stability. However, it does not mean that all the non-tropical features should not be considered. The non-tropical coordinates features have also achieved good performance in applications. In the handwritten digit classification application in Kališnik (2018), the classification rate of tropical coordinates differs by a small amount (less than 1.5% point differences) compared to that of polynomials of Adcock et al. (2016). The higher

prediction rate in Kališnik (2018) could be due to the larger number of features (six features in tropical coordinates vs. four features of polynomials) not the stability of tropical coordinates. As seen in our fingerprint application, the non-tropical coordinates features also achieve high classification rate. Especially, we believe that stability might not be a big concern in our suggested pipeline. We expect that features that are not stable might be automatically removed during the feature selection process. Also, it has not been established theoretically that only stable features guarantee good classification results. It is possible that features could measure variabilities in a transformed persistence diagram, therefore, do not satisfy stability theorem with respect to the Wasserstein or bottleneck distances.

3.5.2 Fingerprint Data Analysis

The assertion in Yager and Amin (2004) that minutiae points are not useful for classification is unfounded and, evidently, untrue. While it is difficult to compare the performance of automated fingerprint classification algorithms across different databases and methodologies, our minutiae-based persistent homology (with a peak accuracy rate of 91.4% in our 3-class setting) appears to perform squarely within the range of accuracies demonstrated by other published fingerprint classification methods. Interestingly, however, the performance precipitously drops when we use only the minutiae locations, rather than the locations and orientations (indeed, our peak accuracy rate there is 62.9%, barely above the baseline comparison rate of 58.4% achieved by always guessing "loop"). From a practical perspective, this reduces the utility of our method since many databases do not include the minutiae orientations and automatically extracting them is an additional non-trivial layer of preprocessing. From a theoretical perspective, however, this prominent role played by the minutiae orientations provides some novel insight into how the global geometry of fingerprint classes influences the local geometry of the ridge pattern—though making this precise remains a challenge.

Turning to our image-based persistent homology approach, we find a peak accuracy rate, 77.1%, that is significantly above the baseline comparison but somewhat short of the state-of-the-art methods in the literature. However, the fact that the only preprocessing this method uses is cropping (not rotating, translating, smoothing, cleaning the image, etc.) means that it still shows significant promise and should be studied further. Moreover, by combining the image-based approach with the minutiae-based approach, our peak accuracy rate increases by 1.7 percentage points (from 91.4% to 93.1%), a small but non-trivial improvement. Intriguingly, in the set of features selected by backwards elimination to achieve this peak accuracy rate, nearly half are image-based (15 out of 32, see Table 3.4) even though in terms of classification (Table 3.2) the minutiae-based features are much stronger than the image-based ones. One possible explanation for this is that the minutiae-based features classify quite well and do the bulk of the work, but to push the rate even higher one needs to add geometric information that is dissimilar to what the minutiae data discern—and while the image-based features accomplish this, each one is so weak that it takes a fairly large number of them to yield a noticeable improvement.

Chapter 4

Statistical Analysis Pipeline for Porous Material Images using Topological Data Analysis

4.1 Background

Algebraic topology offers powerful mathematical tools for describing the connectivity of space, and how the connectivity varies. It connects the local shapes in a dataset to several global connectivity properties, in a concrete, measurable way. Porous materials have been studied using pore geometry and topological characteristics (Scholz et al., 2012; Herring et al., 2013). These approaches focus on analyzing materials by computing Euler characteristics χ, which is the alternating sum of Betti numbers ($\chi = \beta_0 - \beta_1 + \beta_2 - \beta_3 + \cdots$). On the other hand, persistent homology provides a numerical summary of the topological features (connected components, loops, shells, etc.) as a function of a metric. Persistent homology enables researchers to quantify dynamics of topological characteristics and to predict physical behavior of porous materials. It can provide more information than reporting the Betti numbers or Euler characteristics (Robins et al., 2016). In this chapter, we propose a statistical porous materials analysis pipeline using persistent homology.

4.2 Data

We analyze three types of rock data in this chapter. The first is a Focus Ion Beam-Scanning Electron Microscopes (FIB-SEM) dataset of Selma Chalk. The FIB cuts the surface of materials and SEM provides a high resolution image of the surface. The Selma Chalk dataset is based on previously binarized images used in the study by Yoon and Dewers

(2013). Figure 4.1 shows original and binarized images of the Selma Chalk. The original dataset includes a grayscale image that is $930 \times 520 \times 962$ voxels with 15.6 nm resolution ($14.8 \times 8.3 \times 15.3$ microns in size). There are in total six different sizes of subvolumes: 150^3, 300^3, 400^3, 500^3, $600 \times 520 \times 600$, and $765 \times 520 \times \times 765$ voxels. Yoon and Dewers (2013) estimate parameters for each subvolume including porosity, permeability, tortuosity, and anisotropy. Second, we analyze two types of sandstones; Bentheimer and Doddington. The images are $700 \times 700 \times 700$ and $600 \times 600 \times 700$ voxels in size. The geophysical properties for these sandstones are not computed. Last, we analyze sandstone images obtained under nine different stress-strain levels. The stress-strain dataset is generated by simulations. Table 4.1 summarizes the three types of material image data.

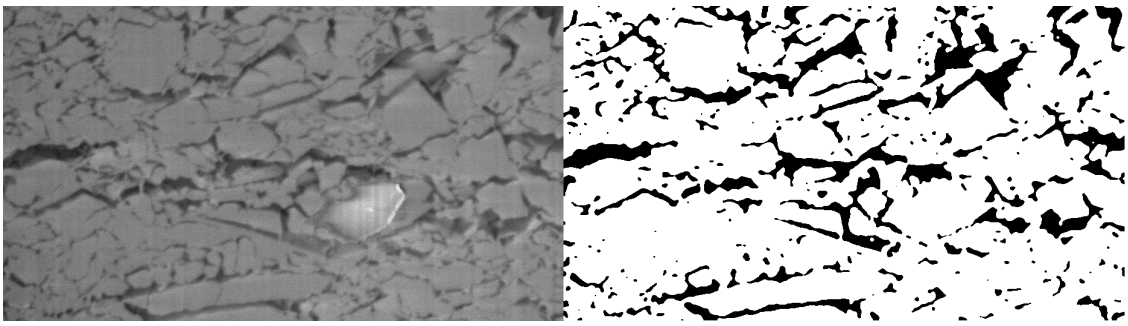


Figure 4.1: Grayscale (left) and binarized (right) image slices of the Selma Chalk by Yoon and Dewers (2013).

Table 4.1: Data summary

Rock type	**Generation**	**Size (pixels)**	**Data type**	**Properties known**
Selma Chalk	FIB-SIM	$930 \times 520 \times 962$	Grayscale, Binary	Porosity, Permeability, Anisotropy, Tortuosity
Bentheimer Doddington	FIB-SIM	$700 \times 700 \times 700$ $600 \times 600 \times 700$	Grayscale	-
Sandstone	Simulations	-	Grayscale	-

4.3 Statistical Analysis Pipeline

In this section, we explain a porous materials analysis pipeline using persistent homology. The analytic scheme is summarized below. Based on the persistent homology computation

result, we suggest methods to determine an appropriate sampling size called the statistical Representative Elementary Volume (sREV) and to predict geophysical properties.

1. Data pre-processing I
    - Original Images → Binary images
2. Persistent homology computation (Robins et al., 2011, 2016)
    - Binary images → Transformed grayscale images → Cubical complexes → Persistence diagrams
3. Data pre-processing II
    - Persistence diagrams → Vectorized persistence diagrams
4. Application I: Sampling
    - Generate different sized sub-volumes → Compute persistent homology → Vectorize persistence diagrams
    - Vectorized persistence diagrams → Similarity metric → Determine sREV
5. Feature extraction
    - Vectorized persistence diagrams → Principal component analysis → Loadings
6. Application II: Modelling
    - Fit a penalized regression model → Prediction
    - Fit SML models → Clustering/classification

4.3.1 Persistent Homology Computation

We use the persistent homology computation framework of Robins et al. (2011, 2016). In Robins et al. (2016), the geometric characteristics of the binary image are defined by the

Signed Euclidean Distance Transform (SEDT). The SEDT assigns a numeric value to each pixel: negative for pore and positive for grain. Its magnitude represents the Euclidean distance between a pixel and the closest opposite status pixel; a large negative value indicates a large pore size, and a large positive value indicates a large grain size. Then, a cubical cell complex is defined based on the discrete Morse function. The SEDT value of a cell is the maximum value of all of its vertices. The cubical cell complex is an appropriate topological space for the images. The components of cubical cell complexes are 0-cell (vertex), 1-cell (edge), 2-cell (patch) and 3-cell (solid). Figure 4.2 shows the four components of a cubical cell complex.

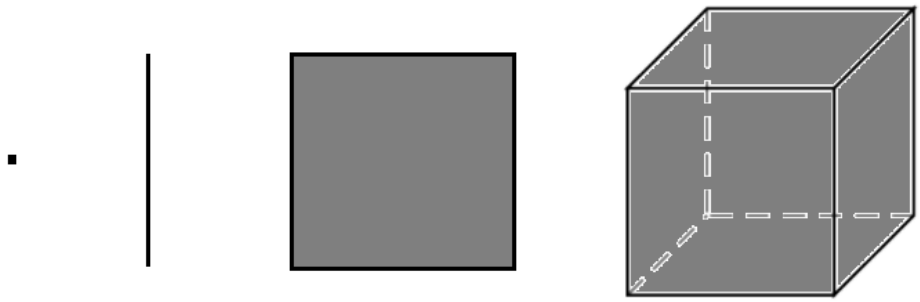


Figure 4.2: Components of a cubical cell complex; 0-cell, 1-cell, 2-cell, and 3-cell, from left to right.

As we change the filtration value, k-cell components are added to the cubical cell complex. That is, we add cells one by one in order of increasing SEDT value, thus a cell is added when the current filtration value reaches the maximum value of all of its vertices. By Morse theory, it is sufficient to track critical points: local minimum (0-cell), local maximum (3-cell), saddle points (1-cell and 2-cell). By tracking the homology of the sequence of cubical complexes, we can compute the persistent homology. An example of sequential changes of cell complexes is shown in Figure 4.3. At first, fhe pixels at the middle of the large pore (large negative SEDT value) first appear in the cell complex. As the filtration moves to the positive direction, pixels in the grain phase are added to the complex. We use the persistent homology computation code Diamorse (Delgado-Friedrichs, 2015).

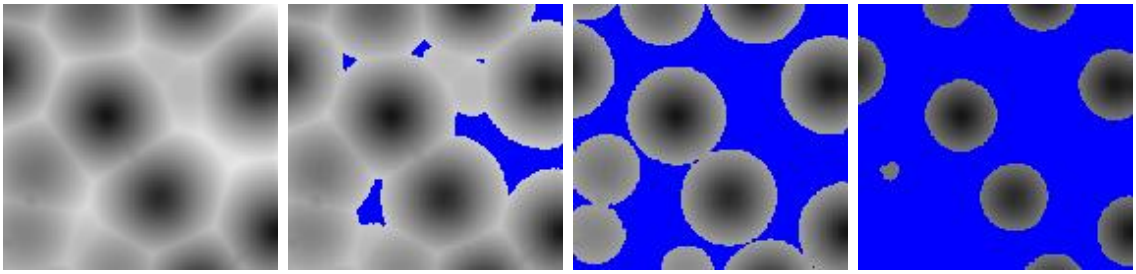


Figure 4.3: SEDT converted image (left) and sequential changes of cubical cell complexes of the images. Blue colored pixels construct the cubical complexes. Modified Figure of Robins et al. (2016).

The persistent homology computation results for each dimension are reported separately: the zero-, one-, and two-dimensional homology groups. Figure 4.4 shows examples of computation results summarized by persistence diagrams. For each dimension's persistence dia-

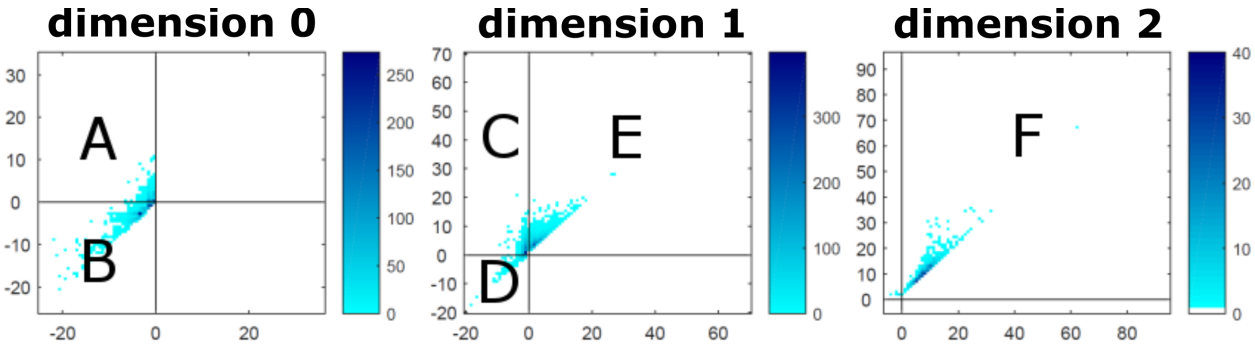


Figure 4.4: Examples of persistence diagrams dimension 0, 1, and 2

gram, the different quadrants of its image reveal different aspects of materials (Robins et al., 2011, 2016). Table 4.2 interprets the regions labeled in Figure 4.4. Visual examples of corresponding structures are given in Figure 2 in the Appendix.

4.3.2 Vectorization of Persistence Diagram

As explained in Chapter 3, classical SML methods cannot be directly applied to the persistence homology computation results; interval data is a non-classical data type and the number of intervals generated varies from dataset to dataset. To overcome the difficulties

Table 4.2: Interpretation of persistence diagrams

Dim.	Region	Structure	Value of X (Birth)	Value of Y (Death)
0	A	Disconnected pore	Size of pore	Narrowest grain contact
	B	Connected pore	Size of pore	Pore throat radius
1	C	Contact grain	Pore tube radius	Grain contact radius
	D	Non-convex pore	Pore tube radius	Non-convex pore throat radius
	E	Non-convex grain	Grain tube radius	Non-convex grain throat radius
2	F	Grain	Grain-contact radius	Size of grain

of directly using intervals, researchers have suggested using vectorized persistence diagrams. There have been different approaches proposed to vectorize persistence diagrams; among these are binning by Bendich et al. (2016) and a persistence image by Adams et al. (2017). The vectorization process has two advantages: 1) existing techniques including image analysis and statistical learning methods can be applied; 2) the mean persistence diagram can be computed. A disadvantage is that the comparison between vectorized persistence diagrams will not be exact, compared to the Wasserstein or bottleneck distances for example.

However, these vectorization methods cannot be directly applied for the persistent homology computation framework of Robins et al. (2016). First, y coordinate transformation used in these methods is not necessary and even makes it difficult to identify structural information. Existing methods transform the mapping of (birth, death) to (birth, death–birth) in persistence diagrams to highlight the distance of the points from the 45-degree line. This is useful when significant features appear in the region far from the 45-degree line. On the other hand, persistence diagrams computed as in Section 4.3.1 records structural characteristics described in Table 4.2. Therefore, the distance from the 45-degree line is not related to the topological significance. Also, such transformation makes it difficult to identify implications of different sections of persistence diagrams for rocks. For example, three quadrants of dimension one persistence diagrams in Figure 4.4 imply different meanings. The transformation maps these quadrants into differently shaped regions. Second, it is not

applicable to vectorize a smoothed persistence diagram. The persistence image suggested in Adams et al. (2017) enables robust transformation by applying a smoothing function on the persistence diagram and converting the smoothed surface into a vector. However, smoothing can make misleading influence over regions which contain different structural information. The boundaries of quadrants in persistence diagrams becomes vague after smoothing, which can lead to a corruption of the underlying characteristics of materials. For example, for dimension one persistence diagrams in Figure 4.4, smoothing can make information of non-convex pores in region D affect and be affected by information of non-convex grains in region E. Even though we apply separate smoothing for each region, an appropriate smoothing has not yet been studied for rocks. It is not known how noise in the original images or filters used in a binarization step affect the persistence diagrams.

We convert persistence diagrams to image vectors similar to Bendich et al. (2016) but without transformation. We bin the elements of the persistence diagram into $m \times m$ bins; each bin is an output pixel. The number of bins m^2 is determined by the number of integers from the floor of minimum to the ceiling of a maximum of the barcodes. In this way, we can avoid having blurred boundaries of the quadrants. Then, we count the number of dots that correspond to barcodes in each bin and assign this as the output pixel intensity. We convert the array of pixels into a vector by scanning the columns: visit the bins in the first column from top to bottom, then the bins in the second column, etc. We found this sufficient, but an alternative would be to scan the image in order of a space-filling curve. Figure 4.5 illustrates the vectorization process.

4.3.3 Determining sREV using Persistence Homology

The three-dimensional material images are expensive data to obtain. For accuracy, we need a sufficiently large subsample; but for efficiency, we do not want to use a larger subsample than necessary. Persistence data depends on the size of the dataset, or subsample of the

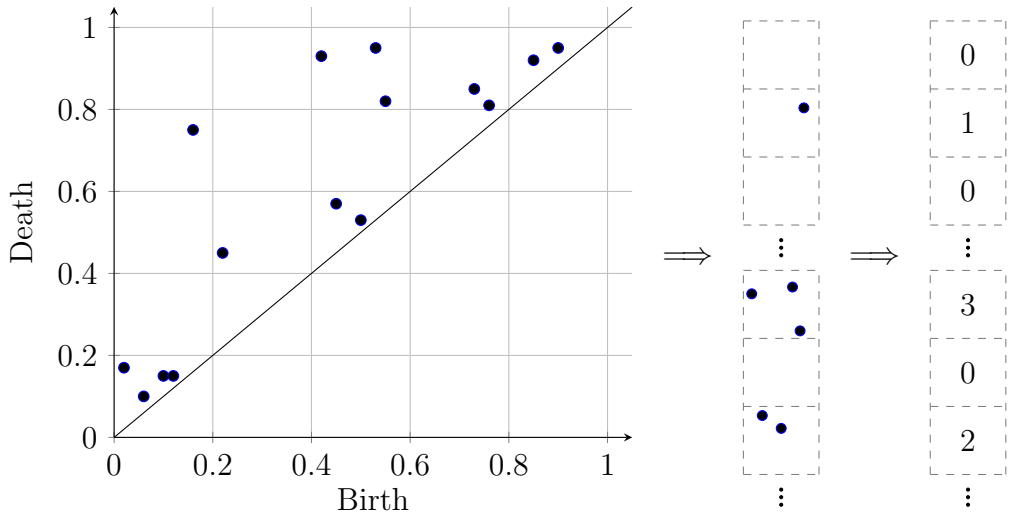


Figure 4.5: Persistence diagram divided by 5×5 pixels (left), concatenated persistence diagram (middle) and vectorized persistence diagram (right)

dataset, being considered. But how it depends is poorly understood, and may change unpredictably as the subsample size increases. Traditional statistical methods use the concept of a representative elementary volume (REV) (Bear, 1972). The REV is the scale at which smaller-scale fluctuations dampen out, and statistically stable properties can be defined. For porous media, such as rocks, REVs for key parameters such as porosity, permeability, and tortuosity enable one to apply continuum methods to predict fluid flow and transport. Also, the REV allows the use of partial differential equations. REVs are thus useful but discard pore-scale information.

The statistical REV, sREV, represents a scale smaller than that of the REV. The sREV is a scale where the means of properties are constant, and their variations are small. The concept of sREV has been used for quantifying microstructures of various materials including single-phase flow in sandstones at the microscale Zhang et al. (2000), mechanical properties of fiber-reinforced composites Trias Mansilla (2005), and transport properties of fuel cell materials Wargo et al. (2012). However, until now, the sREV has not been evaluated for

quantitative analysis of FIB-SEM data with nanopore structures observed in geo-materials: e.g., carbonate rocks and shale mudstones. The sREV is closely related to defining the sampling unit from the rock images, and the "right scale" for rock analysis. However, it is computationally very expensive to determine the sREV. The key properties of materials are computed by running extensive simulations such as the lattice Boltzmann methods (Zhang et al., 2000).

Persistent homology, on the other hand, provides numerical summary of size, structure, and connectivity of materials with relatively low computational load. We suggest the following method to determine the sREV using persistence homology. The underlying hypothesis is: if the structural properties of sampled subvolumes are similar to each other, then persistence diagrams would be similar as well. Therefore, we can determine the sREV by measuring similarity of persistence diagrams.

We propose using similarity measures for images on the vectorized persistence diagrams. Similarity measures for images have been developed to be robust to differences in shift/scale/noise (Wang et al., 2005; Li and Lu, 2009). However, an appropriate measure for persistence diagrams should be sensitive to shift and scale differences while robust to perturbations because there are no rotational or directional changes in persistence diagrams. We consider using measures of mean squared error (MSE) and persistence landscape Bubenik (2015). However, these approaches measure the relative distance between two persistence diagrams. Also, these distances depend on image scales, and so cannot be immediately applied when comparing two subsamples of different sizes. Hence we conclude that these two measures are not the best way to determine sREV for persistent homology.

We instead suggest using the structural similarity (SSIM) of Wang et al. (2004). The SSIM index is defined as a product of three components: the luminance $l(x, y)$, contrast $c(x, y)$, and structure $s(x, y)$. The SSIM varies from -1 to 1, where one indicates two images x and y are identical. Formally, it is defined as

$$\text{SSIM}(x, y) = l(x, y)^\alpha * c(x, y)^\beta * s(x, y)^\gamma$$

where

$$\begin{aligned} l(x, y) &= \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \\ c(x, y) &= \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \\ s(x, y) &= \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}. \end{aligned}$$

Here μ is the average intensity of image and σ is the standard deviation as a measure of intensity spread. We use the default setting $\alpha = \beta = \gamma = 1$ and $C_3 = C_2/2$ as Wang et al. (2004) so that

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}.$$

Instead of computing SSIM for the whole image, Wang et al. (2004) suggest computing SSIM for multiple local blocks of the image. The mean SSIM (MSSIM) is the average of the SSIM values of blocks:

$$\text{MSSIM}(x, y) = \frac{1}{M}\sum_{i=1}^{M}\text{SSIM}(x_i, y_i),$$

where x_i and y_i are the ith block of images x and y.

We compute the MSSIM between vectorized persistence diagrams and their mean image for each subvolume. However, for the persistence diagrams, the MSSIM is very high because most of the image pixels are zero, e.g., all lower diagonal pixels. Therefore, we only consider the local blocks of mean images that have a non-zero element:

$$\text{MSSIM}_{PH}(x, \mu) = \frac{1}{\#\{k|\mu_k \neq 0\}}\sum_{i\in\{k|\mu_k\neq\mathbf{0}\}}\text{SSIM}(x_i, \mu_i).$$

Because there is no standard for deciding sREV using persistent homology, we suggest using the threshold to be 0.9 (weak) and 0.95 (strict) for the MSSIM_{PH}.

4.3.4 Feature Extraction: Principal Component Analysis

We extract features of the vectorized persistence diagrams using principal component analysis (Jolliffe, 2002). First, we subtract the mean (vectorized) persistence diagram from all the persistence diagram vectors. We compute the principal components of the covariance matrix. The principal components form a basis to explain the vectorized persistence diagrams. We use the singular vector decomposition to find the principal components to reduce the computational load. The vectorized persistence diagram can be represented as a linear combination of the principal components.

$$\begin{aligned} &i^{th} \text{ dimension } k \text{ persistence diagram} - \text{dimension } k \text{ mean persistence diagram} \\ &= c_{ik1} * PC_{k1} + c_{ik2} * PC_{k2} + \cdots + c_{ikn} * PC_{kn} \end{aligned}$$

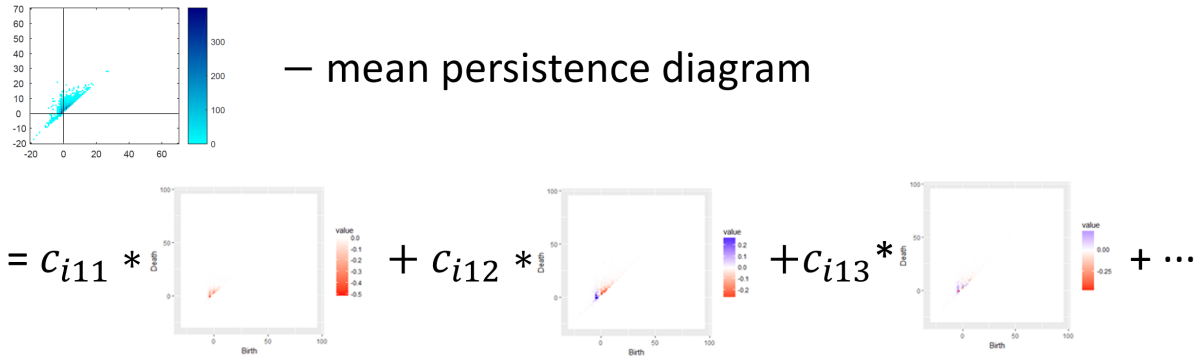


Figure 4.6: Representation of a vectorized persistence diagram as a linear combination of principal components

The coefficients of the principal components are called the *loadings*. We can use the set of loadings to summarize persistence diagrams. The Euclidean vector $v_i = \{c_{i01}, c_{i02}, \cdots c_{ikn}\}$ summarizes the i^{th} porous material. Figure 4.6 illustrates the equation above with the first, second, and third principal components of the dimension one persistence diagram.

Rock Images $\longrightarrow$ Persistence Diagrams $\longrightarrow$ Euclidean Vector of PC Loadings

Once we convert data into the Euclidean vector, we can then apply classical statistical approaches to make an inference. For example, the numeric values can be used as an explanatory variable for classification or regression.

4.3.5 Prediction of Fluid Flow and Transport Characteristics: Penalized Regression Model

We would like to fit a model that explains the geometric properties (y variables) using loadings obtained from the principal component analysis (x variables). If there are n subvolumes, then we will obtain n principal components for each dimension. As a result, the number of loadings obtained for all dimensions is $3n$, and is larger than the number of samples n. This is called the "large p small n" problem.

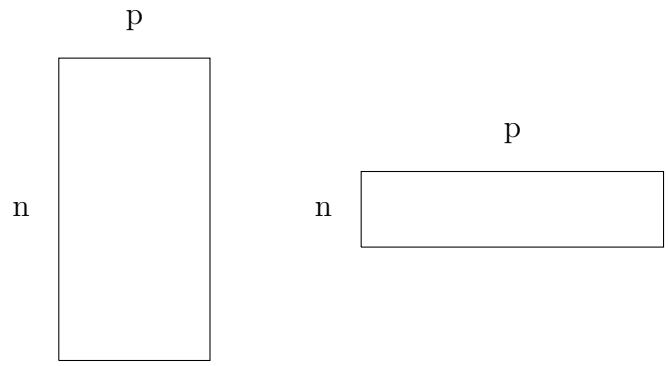


Figure 4.7: "large n small p" vs. "large p small n" data.

One of the solutions to the "large p small n" problem is to use a penalized regression model, an embedded feature selection method. A penalized regression model fits the same linear regression but gives a penalty to the coefficients. The least absolute shrinkage and selection operator (LASSO) is a penalized regression model using the L_1 penalty (Tibshirani, 1996). The result of LASSO can be obtained by solving

$$\min_{\beta}\{||y - X\beta||_2^2 + \delta||\beta||_1\}. \tag{4.1}$$

It fits a regression and does variable selection at the same time. The advantage of LASSO model is that we can see which principal components play a role in predicting fluid flow and

transport properties. As a result, the geophysical variables could be estimated by LASSO model using PCA loadings.

$$y = \text{geophysical variables} \sim f_{LASSO}(\text{PCA loadings}) \tag{4.2}$$

4.4 Results

4.4.1 Application to Stress-strain Data

Stress-strain data are obtained by imposing different stress levels. We use a total of nine different pressure levels: 2%, 4%, 6%, 8%, 10%, 20%, 30%, 40%, and 50% volume decrease from the original rock sample. The corresponding rock images are generated by simulation. We compute the persistent homology of each rock at each stress, summarized as persistence diagrams. The persistence diagrams are given in Figure 3 (dimension 0), 4 (dimension 1), and 5 (dimension 2) of the Appendix. The computed persistence diagrams reflect the structural changes that happen to the rock under stress. First, dimension zero persistence diagrams reflect the decreasing pore size. The negative birth and death values approach to zero as pressure changes from 2% to 50%. The dimension one barcodes represent structural changes in the pore and grain structure. Figure 4.8 shows the percentage changes of dimension one barcodes. The percentages of non-convex pore structures (region D in Figure 4.4) are small, less than 2.5%, but decrease even further as pressure increases. On the other hand, percentages of non-convex structure in the grain phase (region E in Figure 4.4) increase as pressure level changes from 6% to 50%. At pressure level 50%, 90.3% of dimension one barcodes are of non-convex grain structures. This shows that non-convex pore structures and contact grains glue together and form non-convex grain structures as the pressure level increases. Finally, the dimension two persistence diagrams imply that the size of the grains is increasing. As pressure changes from 2% to 50%, the positive pixels in the persistence diagrams tend to shift to the upper-right.

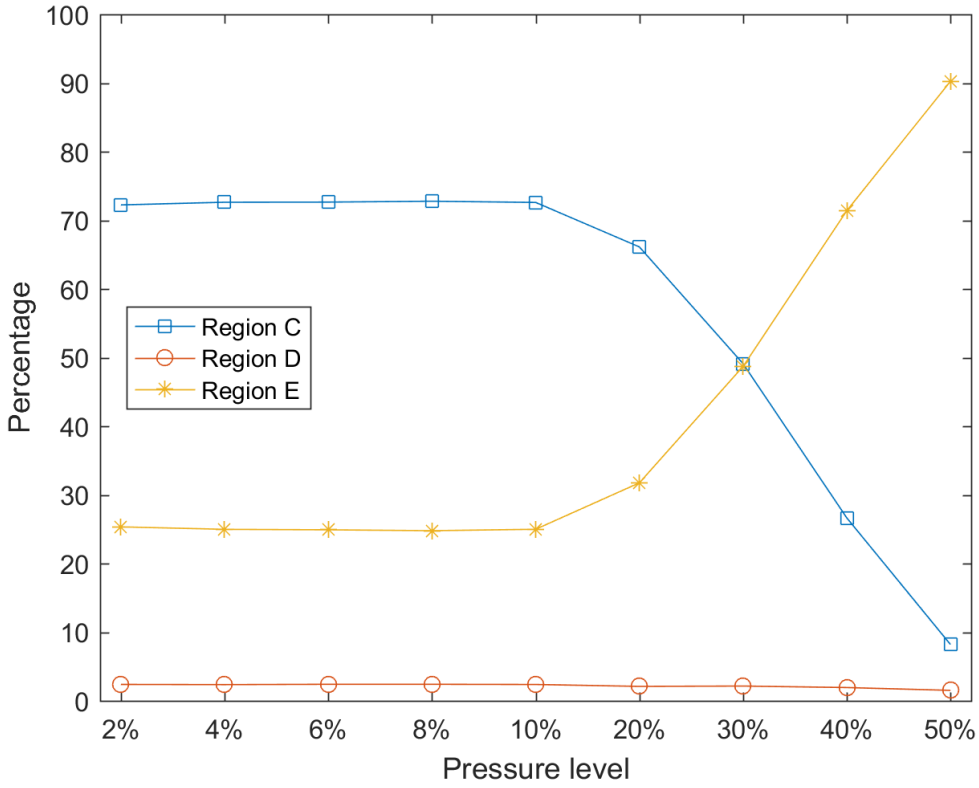


Figure 4.8: Percentages of three regions of dimension one persistence diagram. The labels of three regions correspond to Figure 4.4.

4.4.2 Determination of sREV

We use two datasets for determining sREV; Selma group chalk data of Yoon and Dewers (2013) and Sandstone data (Bentheimer and Doddington).

Selma Group Chalk

Yoon and Dewers (2013) determine sREV of the Selma group chalk data by comparing the variation of five geophysical characteristics; porosity, permeability, tortuosity, anisotropy, and specific surface area. A criterion for sREV scale is set to be the size when the coefficient of variation (the standard deviation divided by the mean) is less than 15% for the five properties.

As a result, the sREV is determined as 400^3 voxels under the weak condition (considering variabilities of porosity, tortuosity, and specific surface area) and $600 \times 520 \times 600$ voxels under the strict condition (considering all the five properties).

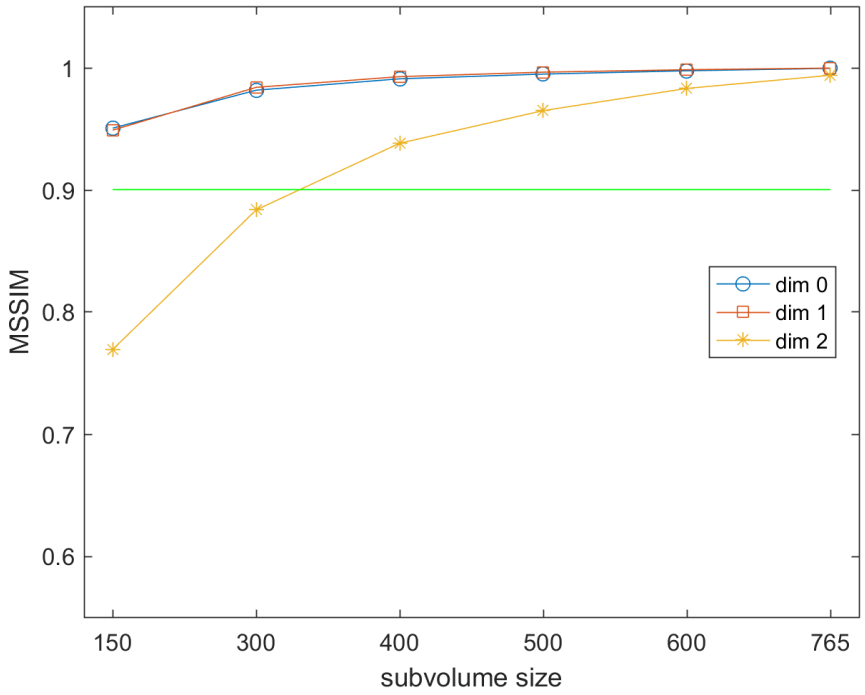


Figure 4.9: Average SSIM (MSSIM_{PH}) of Selma group chalk.

We determine the sREV for the Selma group chalk data as explained in Subsection 4.3.3. Figure 4.9 shows the average SSIM of the mean persistence diagram and persistence diagrams ($\text{MSSIM}_{PH}(x, \mu)$) for six subvolumes. The similarities are measured for persistence diagrams of three dimensions (dimension 0, 1, and 2). As the size of subvolume increases, structures become similar, and the average MSSIM_{PH} values increases. For the Selma group chalk, the biggest differences appear in the dimension 2 persistence diagrams. This implies that the largest variability of Selma group chalk comes from the irregular-sized grains. The average MSSIM_{PH} of dimension 2 exceeds 0.9 and 0.95 at 300^3 and 500^3 subvolume sizes, respectively. Therefore, 300^3 and 500^3 subvolumes could be considered as sREV under the proposed conditions. Whereas Yoon and Dewers (2013) determine the sREV by comparing

the stability of geophysical property values, our persistent homology approach directly compares structural similarities. Also, sREV can be determined with much less computational load by persistent homology.

Two Sandstones

For the Bentheimer and Doddington sandstone data we generate four sizes of subvolumes: 100^3, 150^3, $200 \times 200 \times 150$, and $300 \times 300 \times 200$. The number of subvolumes is 100, 64, 36 and 12, and they are selected to have minimal overlapping regions.

The average SSIM (MSSIM_{PH}) of two sandstones for four subvolumes is computed using persistent homology and Figure 4.10 presents the results. The sandstones show differences compared to the Selma Chalk data. The largest dissimilarity (low MSSIM) for Bentheimer occurs in the dimension 0 persistence diagrams whereas Doddington's appears in the dimension 1 aspects. This implies that the largest structural variability of Bentheimer comes from the differences in size of pore spaces. On the other hand, Doddington has more variability in non-convex pore/grain structures. By applying the same decision rule, the sREV of the Benthheimer is determined to be the size of 150^3 (weak) and $200 \times 200 \times 150$ (strict), and Doddington is determined as size of 150^3 (weak) and $300 \times 300 \times 200$ (strict).

4.4.3 Prediction of Geophysical Properties

We use the Selma group chalk data of Yoon and Dewers (2013). We fit a model only for three smallest sizes of subvolumes (150^3, 300^3, and 400^3) because for the other sizes the number of subvolumes is insufficient. We have 42, 23, and 23 subvolumes for the sizes 150^3, 300^3, and 400^3.

We fit a LASSO model to predict four fluid flow and transport properties: porosity ϕ, permeability k, anisotropy λ, and tortuosity τ. We restate their definitions from Yoon and Dewers (2013) for completeness. Porosity is the ratio of the volume of the pore space over the total volume. Permeability measures how readily a fluid or gas flows through a

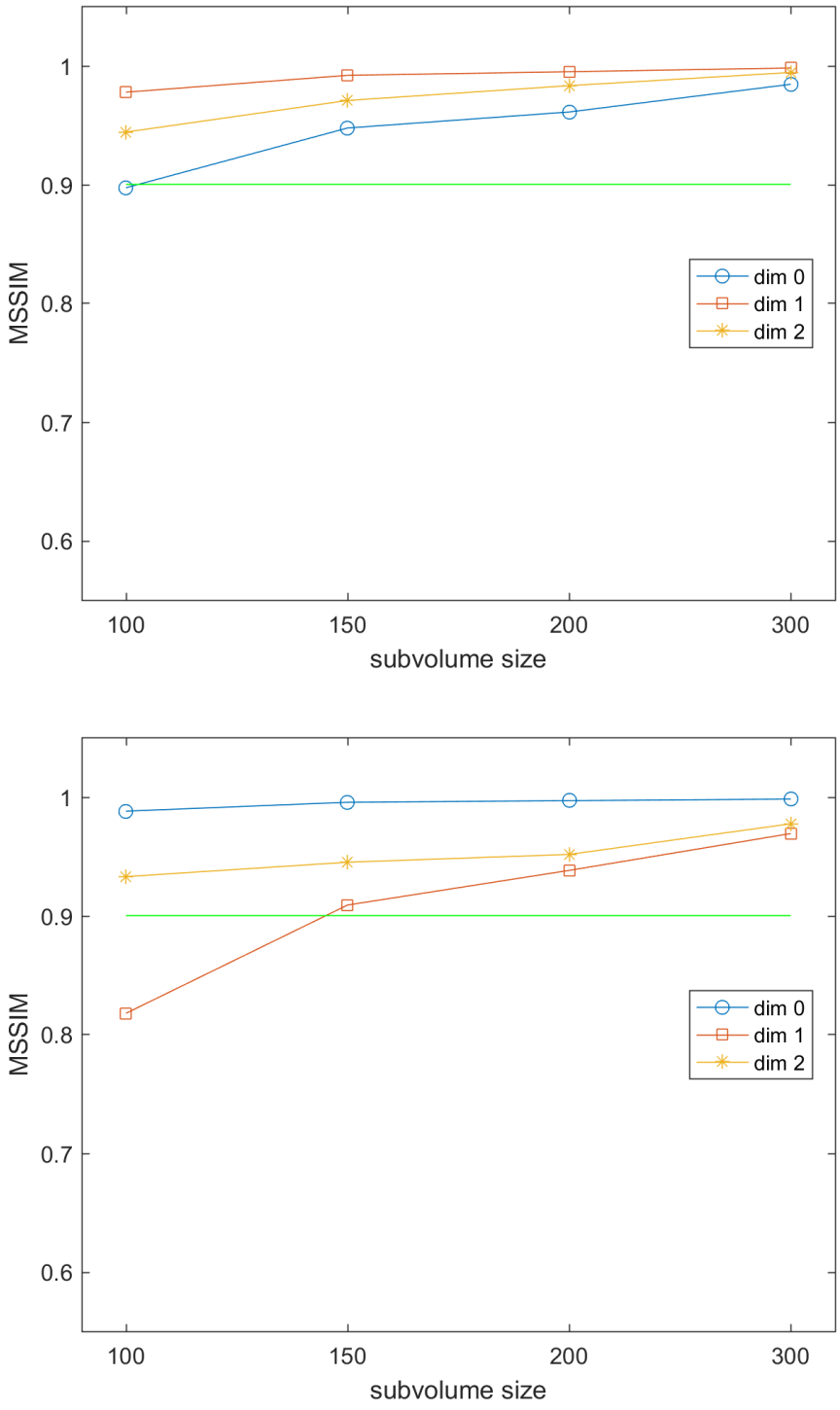


Figure 4.10: Average SSIM (MSSIM_{PH}) of Bentheimer (top) and Doddington (bottom).

material. Permeability is measured in $x, y,$ and z directions. We define the representative permeability as a geometric mean $(k_x * k_y * k_z)^{1/3}$. Anisotropy measures structural differences along the directions. Tortuosity quantifies how much pore paths are twisted. Tortuosity is also measured in three directions. We define the representative tortuosity as an arithmetic mean $(\tau_x + \tau_y + \tau_z)/3$. To decide δ in the LASSO model, we train the model with 3000 repetitions. The ratio of training, validation, and test sets is 60%, 20%, and 20%, respectively.

The prediction results of the four properties are summarized as plots of actual vs. fitted values in Figure 4.11. The black dots represent data points of training and validation, whereas the red dots are the test sets. In an actual vs. fitted plot, the closer dots are to the 45-degree line, the more accurate the prediction. For some geophysical properties, predicted values form straight vertical lines around the total average. This is the case when LASSO drops out loadings (x variables) because they do not contain enough information in predicting the corresponding geophysical variables. The porosity and permeability prediction results show the sudden increase in predictive accuracy at size 400. The anisotropy prediction results show similar predictive accuracy between sizes 300 and 400. At the subvolume size of 400, predictions become accurate for three fluid flow and transport variables. This corresponds to our expectation as it is the sREV size found in Yoon and Dewers (2013).

4.5 Discussion

For two sandstones Bentheimer and Doddington, we also generate subvolumes of size 50^3 as the smallest subvolume. However, the MSSIM_{PH} values of size 50^3 are higher than the larger-sized subvolumes. When the subvolume size is too small, then it does not include much grain or pore structures. As a result, only a few barcodes are generated for the small-sized subvolumes. In terms of MSSIM_{PH}, differences between vectorized persistence diagrams are not that large.

We also face a similar problem in Rotleigend sandstone data sREV determination, which is presented in Figure 6 in Appendix. We first generate five subvolumes 50^3, 100^3, 150^3,

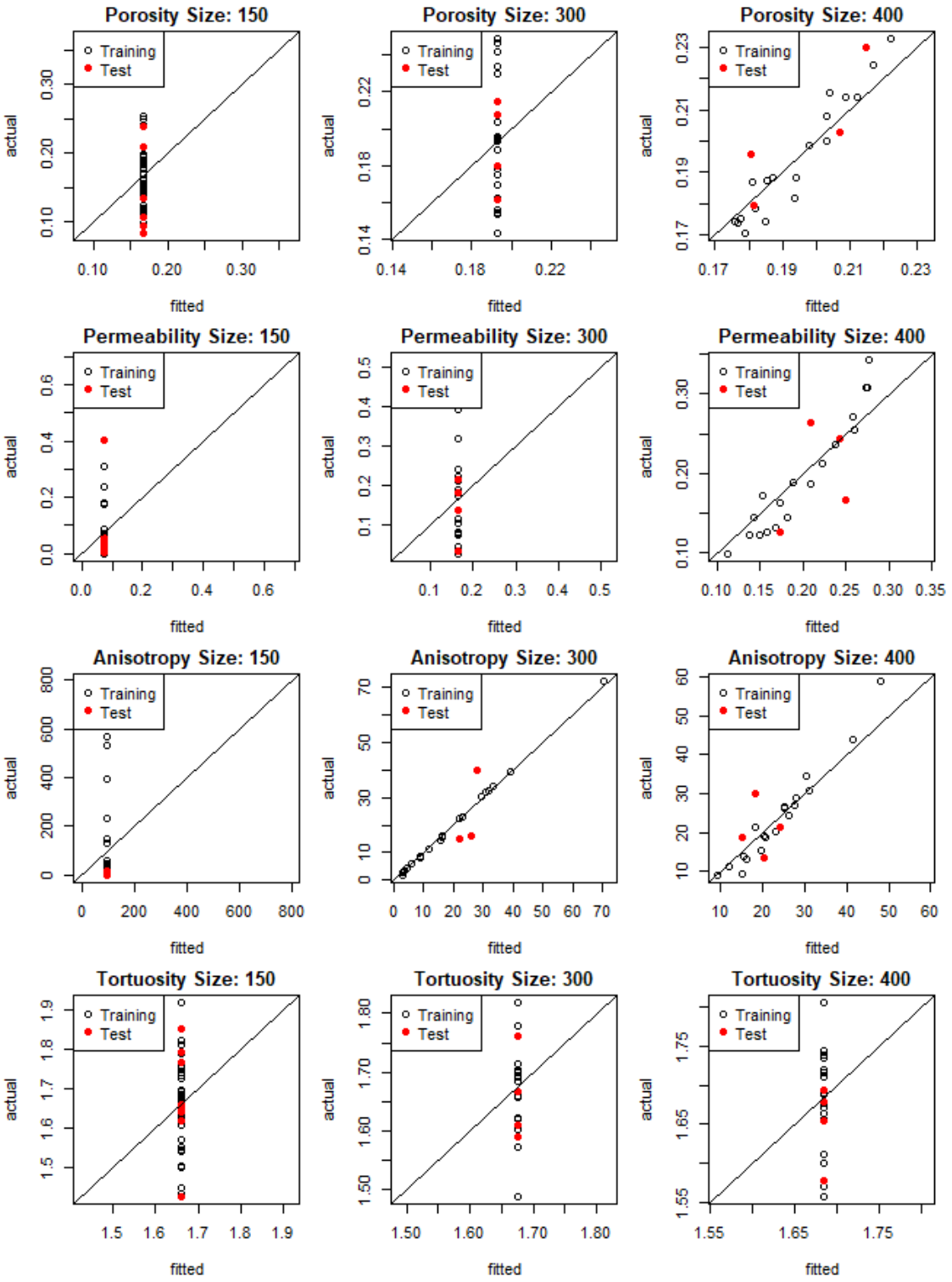


Figure 4.11: Fitted vs. actual plots of porosity (first row), permeability (second row), anisotropy (third row), and tortuosity (last row) at size 150^3 (first column), 300^3 (second column), and 400^3 (last column). Points closer to the 45 degree line imply more accurate prediction results.

$200 \times 200 \times 150$, and $300 \times 300 \times 200$, same as other sandstones and compute MSSIM_{PH}. However, in some of the 50^3 subvolumes no dimension zero barcodes are generated, which implies that the corresponding subvolumes are placed inside of one of the sandstone grains. Although the Rotleigend image includes more pixels ($700 \times 700 \times 980$) than other sandstone data, they are obtained in a higher resolution and smaller size. According to Figure 6, all the MSSIM_{PH} values are greater than 0.95, so sREV would be determined to be 100^3. However, this is due to the small-sized subvolumes, not the structural similarity. We expect the MSSIM_{PH} graph to be somewhat similar to v-shaped according to the size of subvolumes (high MSSIM_{PH} values for very small sized- and large sized-subvolumes, and lower MSSIM_{PH} values in the middle). The sREV can be selected to be the size of subvolume that exceeds 0.9 or 0.95 for the second time in that case. Also, for the consistency of analysis, we recommend selecting subvolume size considering the resolution of images and size of materials, not just the number of pixels.

Chapter 5

Conclusion and Future Research Directions

TDA is a relatively new approach that has revealed different aspects of data in terms of their shape and structure. Our research focuses on providing better statistical inference and modeling for persistent homology and extending its application boundaries.

First, we propose a novel topological summary plot, called a persistence terrace for point cloud data. The persistence terrace incorporates a wide range of smoothing parameters similar to a scale-space analysis and is robust, multi-scale, and parameter-free. This plot allows one to isolate distinct topological signals that may have merged for any fixed value of the smoothing parameter, and it also allows one to infer the size and point density of the topological features.

Second, we introduce an improved method for persistent homology to be used in SML approaches. We suggest a large set of features that reflect variations between sets of barcodes along with their implications, and feature selection methods for SML modeling. The proposed method is applied to fingerprint classification and achieves near state-of-the-art classification accuracy rates by applying it to 3-dimensional point clouds of oriented minutiae points and fingerprint ink-roll images. The suggested approach allows us to explore feature selection on barcodes, an important topic at the interface between persistent homology and SML methods.

Third, we propose a porous materials analysis pipeline using persistent homology. We first compute persistent homology of binarized 3D images of sampled material subvolumes. We convert persistence diagrams into image vectors to analyze the similarity of the homology of the material images using the mature tools for image analysis. Each image is treated as

a vector, and we compute its principal components to extract features. We fit a statistical model using the loadings of principal components to estimate material porosity, permeability, anisotropy, and tortuosity. We also propose an adaptive version of the structural similarity index as a measure to determine the statistical representative elementary volumes. Thus we provide a capability for making a statistical inference of the fluid flow and transport properties of porous materials based on their geometry and connectivity.

In the following subsections, we present future research topics and directions.

5.1 Improvement of Persistence Terrace as an Inference Tool

In the persistence terrace, topological features in point cloud data are represented as terrace layers and we can make a robust estimation of the feature's size and point density. However, when there is a large number of features, it is difficult to disentangle the individual layers. We recommend the terrace area plot as an aid for determining the number of significant features, but one still needs to pay attention to the terrace layers to make an accurate estimate; even so, there will inevitably be differing interpretations of the persistence terrace. In a future paper, we plan to develop a method to label the different layers systematically to help analyze topologically complicated data.

5.2 Interface and Fingerprint Application

5.2.1 Additional Features

Regression coefficients feature suggested in Subsection 3.2.2 are most beneficial if it is used for dimension 0 of direct estimation approach. In most dimensions, birth points of barcodes are not zero. If only endpoints are used, then the information contained in intervals would be lost. Instead, we can fit a polynomial regression model on the transformed persistence diagram $(x_i, y_i - x_i)$.

In case of low signal-to-noise ratio data, a larger number of short-length bars would be generated. A polynomial regression model on the transformed persistence diagram might not reflect a small number of long-length (significant) bars. In the future research, we aim to suggest different summary of persistent homology results called "persistence process". Let $X(t)$ be the persistence process at filtration t where $X(0) = 0$. For each bar (x_i, y_i), $X(t)$ increases $w_i \times (y_i - x_i)$ at $t = x_i$ and decreases the same amount at $t = y_i$. The weight w_i can be determined by how much importance to be assigned to longer-length intervals. The persistence process converts a set of barcodes into a single line. There is a one-to-one correspondence between the persistence process and barcodes. The persistence image suggested in Adams et al. (2017) converts persistence diagram into three-dimensional space using a smoothing function. The persistence process converts barcodes into points/lines on two-dimensional space without a smoothing function. We can fit the piecewise polynomial (spline) and use their coefficients to compare the differences between a multiset of barcodes.

We may describe the barcodes using probability distributions. For example, we may estimate a rate parameter λ of exponential distribution on $x_i * (y_i - x_i)$ or shape and rate parameters of Erlang distribution.

5.2.2 Feature Selection Methods

The forward/backward elimination which implements a greedy algorithm, is a computationally expensive method. We may use feature selection embedded SML methods, which performs feature selection and classification at the same time. One example of the embedded SML methods is a sparse method. There have been different approaches including sparse discriminant analysis (Clemmensen et al., 2011; Gaynanova and Kolar, 2015) and sparse support vector machine (Bradley and Mangasarian, 1998; Zhu et al., 2003). We plan to investigate the sparse methods for barcode feature selection and compare the classification/prediction results.

5.2.3 Fingerprint Data Application

Since directly inferring geometric structure from complicated barcodes is not plausible, we base our persistent homology approach on supervised learning; consequently, our results are strongly influenced by the size of the data set. We want to use both oriented minutiae points and matching JPEG ink-roll images, and this narrows our choice of data set down to NIST SD-27, which is rather small compared to most fingerprint databases used in the fingerprint community for training and testing purposes. In particular, we only have 13 arches, which makes "learning" the shape of their barcodes quite challenging (and indeed the confusion matrices in Table 3.3 reveal that arches have a vastly greater frequency of misclassification than the other two classes).

Persistent homology is essentially invariant under translation and rotation—except that our image-based method involves the choice of orthogonal directions to slant the surface, but we do not believe the results would be very sensitive to small changes in these directions—so there is little challenge to using our methodology across data sets, and no challenge at all to doing so for the minutiae-based approach, and doing so should significantly improve accuracy rates. Practically speaking, we believe that one could, for instance, train a minutiae-based persistent homology classifier on large data sets and then apply it to any new data set that contains minutiae point locations and orientations, no matter how they are recorded.

More generally, we believe that persistent homology provides an important new tool for attacking difficult pattern analysis problems such as fingerprint classification. While much remains to be understood regarding the interface between persistent homology and machine learning, we hope that this study helps provide some insight into feature selection on barcodes—both by summarizing and introducing a convenient collection of features and by exploring their impact on a concrete, well-studied classification problem.

We suggest some topics for future work: (1) training and testing the method introduced in Section 3.4 on a different fingerprint database, or a larger set of features, to see how much the accuracy can be improved; (2) incorporating more sophisticated techniques from statistics

and machine learning, since there is likely room here as well for significant improvement; (3) extend the methods by applying them to different types of minutiae (a single line and multiple lines); (4) adapting and applying persistent homology methods to other problems in fingerprint analysis, such as matching noisy latent prints.

5.3 Porous Material Analysis

5.3.1 Constructing Larger Database

Persistent homology provides information of structure and connectivity, and it could be used as a "fingerprint" of materials. However, we only analyze a few types of rocks and porous materials in Chapter 4. It would be worthwhile to build a database of "fingerprint" of different types of materials using persistent homology.

5.3.2 Relationship with Rock Physics

We successfully estimate key geophysical variables using features extracted from persistence diagrams. However, it has not been studied why those features have such prediction power. We aim to further investigate a relationship between persistent homology results and rock physics. First, we plan to study how the features selected by a penalized regression model characterize actual rocks physics by collaborating with geologists. Also, our study assumes that noise on the 3D material images will be removed in a binarization process. However, it is not realistic to assume that all the measurement error is eliminated. In future research, we plan to define types of possible noise/errors on images and examine how they affect persistence diagrams; noise makes vertical or horizontal shift points on persistence diagrams or their regions. Such information can be further utilized to define an appropriate smoothing function for persistence diagram vectorization or to make comparisons.

5.3.3 Prediction Models and Extensions to Clustering/classification

We plan to implement prediction models that can make further statistical inferences, such as producing confidence intervals. For the SML methods, a vector of loadings computed by applying principal component analysis to the vectorized persistence diagrams can be used as an input object. However, we did not have enough rock types to study this. For future work, if data on multiple rock types are given, we could attempt to classify/cluster them. Also, it would be worthwhile to study how to apply SML methods to the persistence diagrams best, expanding our normalization and scaling rigorously.

Bibliography

Adams, H., T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, and L. Ziegelmeier (2017). Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research 18*, 1–35.

Adcock, A., E. Carlsson, and G. Carlsson (2016). The ring of algebraic functions on persistence bar codes. *Homology, Homotopy and Applications 18*, 381–402.

Ahmad, F. and D. Mohamed (2009). A review on fingerprint classification techniques. *International Conference on Computer Technology and Development 2*, 411–415.

Bear, J. (1972). *Dynamics of Fluids in Porous Media*. Dover Civil and Mechanical Engineering Series. Dover.

Bendich, P., S. P. Chin, J. Clark, J. Desena, J. Harer, E. Munch, A. Newman, D. Porter, D. Rouse, N. Strawn, and A. Watkins (2016). Topological and statistical behavior classifiers for tracking applications. *IEEE Transactions on Aerospace and Electronic Systems 52*, 2644–2661.

Bendich, P., J. S. Marron, E. Miller, A. Pieloch, and S. Skwerer (2016). Persistent homology analysis of brain artery trees. *Annals of Applied Statistics 10*, 198–218.

Billard, L. and E. Diday (2007). *Symbolic Data Analysis : Conceptual Statistics and Data Mining*. Hoboken, NJ: John Wiley & Sons Inc.

Bobrowski, O., S. Mukherjee, and J. E. Taylor (2017). Topological consistency via kernel estimation. *Bernoulli 23*, 288–328.

Borsuk, K. (1948). On the imbedding of systems of compacta in simplicial complexes. *Polska Akademia Nauk. Fundamenta Mathematicae 35*, 217–234.

Bradley, P. S. and O. L. Mangasarian (1998). Feature selection via concave minimization and support vector machines. *Proceedings of the Fifteenth International Conference on Machine Learning*, 82–90.

Bubenik, P. (2015). Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research 16*, 77–102.

Carlsson, G. (2009). Topology and data. *Bulletin of the American Mathematical Society 46*, 255–308.

Carlsson, G. and S. K. Verovšek (2016). Symmetric and r-symmetric tropical polynomials and rational functions. *Journal of Pure and Applied Algebra 220*, 3610 – 3627.

Chan, J. M., G. Carlsson, and R. Rabadan (2013). Topology of viral evolution. *Proceedings of the National Academy of Sciences of the United States of America 110*, 18566–18571.

Chaudhuri, P. and J. S. Marron (1999). Sizer for exploration of structures in curves. *Journal of the American Statistical Association 94*, 807–823.

Chaudhuri, P. and J. S. Marron (2000). Scale space view of curve estimation. *The Annals of Statistics 28*, 408–428.

Chazal, F., D. Cohen-Steiner, and Q. Mérigot (2011). Geometric inference for probability measures. *Foundations of Computational Mathematics 11*, 733–751.

Chazal, F., V. de Silva, M. Glisse, and S. Oudot (2012). The structure and stability of persistence modules. unpublished manuscript, available at arXiv:1207.3674.

Chazal, F., B. Fasy, F. Lecci, B. Michel, A. Rinaldo, and L. Wasserman (2017). Robust topological inference: Distance to a measure and kernel distance. *Journal of Machine Learning Research*. To appear.

Chazal, F., B. T. Fasy, F. Lecci, A. Rinaldo, and L. Wasserman (2015). Stochastic convergence of persistence landscapes and silhouettes. *Journal of Computational Geometry 6*, 140–161.

Chazal, F. and B. Michel (2017). An introduction to topological data analysis: fundamental and practical aspects for data scientists. unpublished manuscript, available at arXiv:1710.04019.

Clemmensen, L., T. Hastie, D. Witten, and B. Ersbll (2011). Sparse discriminant analysis. *Technometrics 53*, 406–413.

Cohen-Steiner, D., H. Edelsbrunner, and J. Harer (2007). Stability of persistence diagrams. *Discrete & Computational Geometry 37*, 103–120.

Cohen-Steiner, D., H. Edelsbrunner, J. Harer, and Y. Mileyko (2010). Lipschitz functions have l p -stable persistence. *Foundations of Computational Mathematics 10*, 127–139.

Delgado-Friedrichs, O. (2015). *Diamorse: Digital image analysis using discrete Morse theory and persistent homology.* `https://github.com/AppliedMathematicsANU/diamorse`.

Edelsbrunner, H. and J. Harer (2008). Persistent homology - a survey. *Surveys on Discrete and Computational Geometry: Twenty Years Later 453*, 257–282.

Edelsbrunner, H. and J. Harer (2010). *Computational Topology : An Introduction.* Providence, RI: American Mathematical Society.

Fan, J. and Y. Fan (2008). High-dimensional classification using features annealed independence rules. *The Annals of Statistics 36*, 2605–2637.

Fasy, B. T., J. Kim, F. Lecci, and C. Maria (2014). *Introduction to the R package TDA.* unpublished manuscript, available at arXiv:1411.1830.

Fasy, B. T., F. Lecci, A. Rinaldo, L. Wasserman, S. Balakrishnan, and A. Singh (2014). Confidence sets for persistence diagrams. *Annals of Statistics 42*, 2301–2339.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics 7*, 179–188.

Galton, F. (1892). *Finger Prints*. McMillan, London.

Gameiro, M., Y. Hiraoka, S. Izumi, M. Kramar, K. Mischaikow, and V. Nanda (2015). A topological measurement of protein compressibility. *Japan Journal of Industrial and Applied Mathematics 32*, 1–17.

Gaynanova, I. and M. Kolar (2015). Optimal variable selection in multi-group sparse discriminant analysis. *Electronic Journal of Statistics 9*, 2007–2034.

Geisser, S. (1975). The predictive sample reuse method with applications. *Journal of the American Statistical Association 70*, 320–328.

Ghrist, R. (2008). Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society 45*, 61–75.

Guyon, I. and A. Elisseeff (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research 3*, 1157–1182.

Hall, M. A. (1999). Correlation-based feature selection for machine learning. *PhD dissertation, Department of Computer Science, University of Waikato*.

Hatcher, A. (2002). *Algebraic Topology*. New York, NY: Cambridge University Press.

Heo, G., J. Gamble, and P. T. Kim (2012). Topological analysis of variance and the maxillary complex. *Journal of the American Statistical Association 107*, 477–492.

Herring, A. L., E. J. Harper, L. Andersson, A. Sheppard, B. K. Bay, and D. Wildenschild (2013). Effect of fluid topology on residual nonwetting phase trapping: Implications for geologic co2 sequestration. *Advances in Water Resources 62*, 47 – 58.

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013). *An Introduction to Statistical Learning: With Applications in R.* New York, NY: Springer.

Jolliffe, I. (2002). *Principal Component Analysis.* Second Edition. Springer Series in Statistics. Springer.

Kališnik, S. (2018). Tropical coordinates on the space of persistence barcodes. *Foundations of Computational Mathematics.* DOI: https://doi.org/10.1007/s10208-018-9379-y.

Kious, W. and R. Tilling (1996). *This Dynamic Earth: The Story of Plate Tectonics.* U.S. Geological Survey.

Kusano, G., K. Fukumizu, and Y. Hiraoka (2016). Persistence weighted gaussian kernel for topological data analysis. *Proceedings of the 33rd International Conference on International Conference on Machine Learning 48*, 2004–2013.

Kwitt, R., S. Huber, M. Niethammer, W. Lin, and U. Bauer (2015). Statistical topological data analysis - a kernel perspective. *Advances in Neural Information Processing Systems 28*, 3070–3078.

Lee, H., M. Chung, H. Kang, B. Kim, and D. Lee (2011). Discriminative persistent homology of brain networks. *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 841–844.

Lee, H., H. Kang, M. K. Chung, B. N. Kim, and D. S. Lee (2012). Persistent brain network homology from the perspective of dendrogram. *IEEE Transactions on Medical Imaging 31*, 2267–2277.

Li, J., K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu (2017). Feature selection: A data perspective. *ACM Computing Surveys*, 1–45.

Li, J. and B.-L. Lu (2009). An adaptive image euclidean distance. *Pattern Recognition 42*, 349–357.

Lum, P. Y., G. Singh, A. Lehman, T. Ishkanov, M. Vejdemo-Johansson, M. Alagappan, J. Carlsson, and G. Carlsson (2013). Extracting insights from the shape of complex data using topology. *Scientific Reports 3*, 1236. DOI: http://dx.doi.org/10.1038/srep01236.

Maclagan, D. and B. Sturmfels (2015). *Introduction to Tropical Geometry*, Volume 161. American Mathematical Society.

Maltoni, D., D. Maio, A. Jain, and S. Prabhakar (2009). *Handbook of Fingerprint Recognition*. Springer, London.

Marron, J. S. and A. M. Alonso (2014). Overview of object oriented data analysis. *Biometrical Journal 56*, 732–753.

MATLAB (2016). *version 9.10.0 (R2016b)*. Natick, MA: The MathWorks Inc.

Miller, A. J. (1990). *Subset Selection in Regression*. UK: Chapman and Hall.

Mula, J., J. D. Lee, F. Liu, L. Yang, and C. A. Peterson (2013). Automated image analysis of skeletal muscle fiber cross-sectional area. *Journal of Applied Physiology 114*, 148–155.

Munch, E., K. Turner, P. Bendich, S. Mukherjee, J. Mattingly, and J. Harer (2015). Probabilistic frchet means for time varying persistence diagrams. *Electronic Journal of Statistics 9*, 1173–1204.

Nicolau, M., A. J. Levine, and G. Carlsson (2011). Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences of the United States of America 108*, 7265–7270.

Petri, G., P. Expert, F. Turkheimer, R. Carhart-Harris, D. Nutt, P. J. Hellyer, and F. Vaccarino (2014). Homological scaffolds of brain functional networks. *Journal of The Royal Society Interface 11*. DOI: http://dx.doi.org/10.1098/rsif.2014.0873.

Phillips, J. M., B. Wang, and Y. Zheng (2013). Geometric inference on kernel density estimates. unpublished manuscript, available at arXiv:1307.7760.

Picard, R. R. and R. D. Cook (1984). Cross-validation of regression models. *Journal of the American Statistical Association 79*, 575–583.

Ramsay, J. and B. W. Silverman (2005). *Functional Data Analysis*. New York, NY: Springer-Verlag.

Robins, V., M. Saadatfar, O. Delgado-Friedrichs, and A. P. Sheppard (2016). Percolating length scales from topological persistence analysis of micro-ct images of porous materials. *Water Resources Research 52*, 315–329.

Robins, V. and K. Turner (2016). Principal component analysis of persistent homology rank functions with case studies of spatial point patterns, sphere packing and colloids. *Physica D: Nonlinear Phenomena 334*, 99–117.

Robins, V., P. J. Wood, and A. P. Sheppard (2011). Theory and algorithms for constructing discrete morse complexes from grayscale digital images. *IEEE Transactions on pattern analysis and machine intelligence 33*, 1646–1658.

Saadatfar, M., H. Takeuchi, V. Robins, N. Francois, and Y. Hiraoka (2017). Pore configuration landscape of granular crystallization. *Nature Communications 8*. DOI: http://dx.doi.org/10.1038/ncomms15082.

Scholz, C., F. Wirner, J. Götz, U. Rüde, G. E. Schröder-Turk, K. Mecke, and C. Bechinger (2012). Permeability of porous materials determined from the euler characteristic. *Physical Review Letters 109*, 264504.

Speyer, D. and B. Sturmfels (2009). Tropical mathematics. *Mathematics Magazine 82*, 163–173.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological) 36*, 111–147.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological) 58*, 267–288.

Trias Mansilla, D. (2005). *Analysis and Simulation of Transverse Random Fracture of Long Fibre Reinforced Composites*. University of Girona, Spain.

Turner, K., S. Mukherjee, and D. M. Boyer (2014). Persistent homology transform for modeling shapes and surfaces. *Information and Inference: A Journal of the IMA 3*, 310–344.

Wang, L., Y. Zhang, and J. Feng (2005). On the euclidean distance of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 1334–1339.

Wang, Z., A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing 13*, 600–612.

Wargo, E., A. Hanna, A. een, S. Kalidindi, and E. Kumbur (2012). Selection of representative volume elements for pore-scale analysis of transport in fuel cell materials. *Journal of Power Sources 197*, 168–179.

Wilson, C., G. Candela, and C. Watson (1993). Neural network fingerprint classification. *Journal of Artificial Neural Networks 1*, 203–228.

Yager, N. and A. Amin (2004). Fingerprint classification: a review. *Pattern Analysis and Applications 7*, 77–93.

Yoon, H. and T. A. Dewers (2013). Nanopore structures, statistically representative elementary volumes, and transport properties of chalk. *Geophysical Research Letters 40*, 4294–4298.

Zhang, D., R. Zhang, S. Chen, and W. E. Soll (2000). Pore scale study of flow in porous media: Scale dependency, REV, and statistical REV. *Geophysical Research Letters 27*, 1195–1198.

Zhu, J., S. Rosset, T. Hastie, and R. Tibshirani (2003). 1-norm support vector machines. *Proceedings of the 16th International Conference on Neural Information Processing Systems*, 49–56.

Zomorodian, A. (2012). Topological data analysis. *Advances in Applied and Computational Topology 70*, 1–39.

Zomorodian, A. and G. Carlsson (2005). Computing persistent homology. *Discrete & Computational Geometry 33*, 249–274.

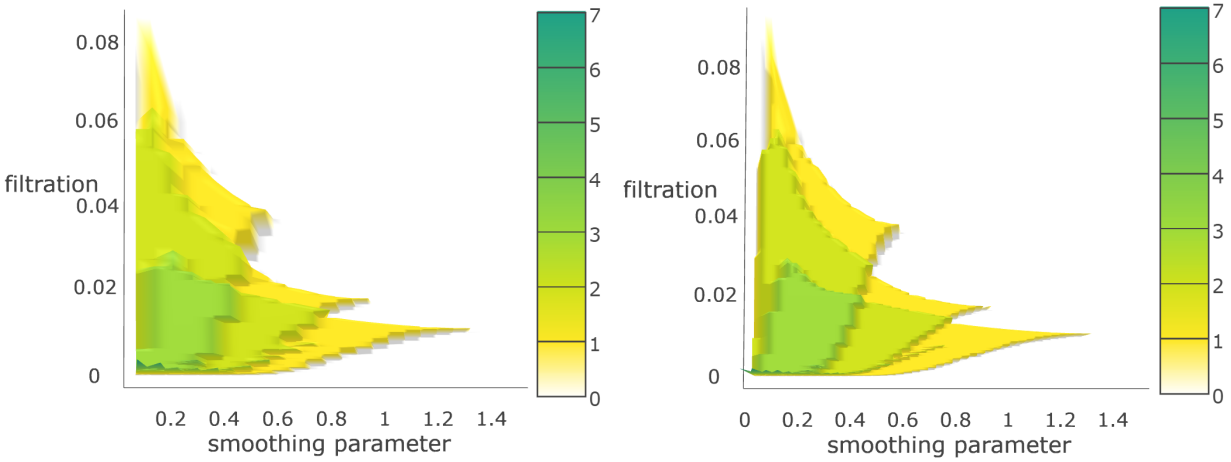
Appendix

Algorithm 1 Compute Betti number values and locations from barcodes

Input *sp* (smoothing parameter vector) and *barcodes* (sets of barcodes)
kholes ← NULL (List of Betti number values and locations, for each smoothing parameter value)
for $i = 1 \rightarrow$ length(*sp*) **do**
  kbarcode ← *barcodes*[[*i*]]{ dim = *k*, birth, death } (Select *k*-dimensional barcode obtained from *i*th smoothing parameter value)
  m ← column length of kbarcode (Number of *k*-dimensional bars)
  if $m = 0$ (No *k*-dimensional bars) **then**
    track ← { filtration=0, numk= 0}
  else
    filtration ← { birth values in *kbarcode*; death values in *kbarcode* }
    kBetti ← { $\mathbf{1_m}$;$\mathbf{-1_m}$ }
    track ← { filtration, kBetti } (Bind two vectors into ($2m \times 2$) matrix, thereby assigning 1 and -1 to the birth and death points, respectively)
    Sort *track* matrix by 'filtration' in ascending order
    track$kBetti ← cumsum(*track*$kBetti) (Replace the kBetti column by its cumulative sum)
  end if
  kholes[[*i*]] ← *track*
end for
return *kholes*

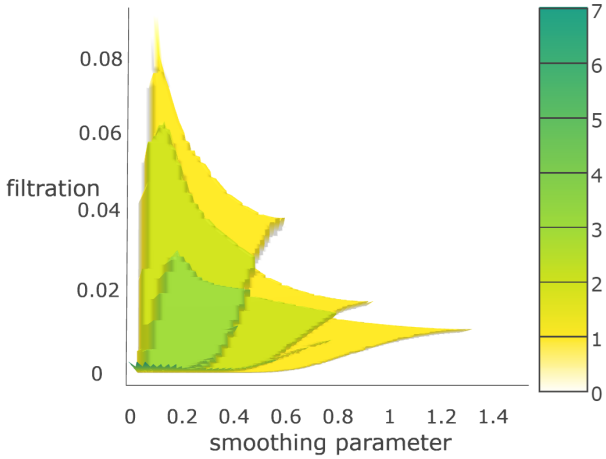
Algorithm 2 Compute persistence terrace matrix from Betti number values/locations

Input sp (smoothing parameter vector) and $kholes = \{$ filtration, kth Betti number $\}$
$n \leftarrow \text{length}(sp)$
$xvec \leftarrow$ sp (x values vector: smoothing parameters)
$yvec \leftarrow$ NULL (y values vector: filtration values when the kth Betti number change)
for $i = 1 \rightarrow n$ **do**
  $yvec \leftarrow \{\ yvec,\ kholes[[i]]\$\text{filtration}\ \}$ (Stack all filtration values)
end for
$yvec \leftarrow \text{sort}(yvec)$ (Sort $yvec$ in descending order)
$zmat \leftarrow \mathbf{0}_{\mathbf{length(xvec)} \times \mathbf{length(yvec)}}$ (z values matrix: kth Betti number)
for $p = 1 \rightarrow n$ **do**
  $filtration \leftarrow kholes[[p]]\$filtration$
  $kBetti \leftarrow kholes[[p]]\$kBetti$
  $zvec \leftarrow \mathbf{0}_{\mathbf{length(yvec)}}$
  for $q = 1 \rightarrow \text{length}(kBetti)$ **do**
    $zvec = zvec + (filtration[q+1] < yvec) * (yvec \leq filtration[q]) * kBetti[q]$ (Fill out kth Betti numbers for all filtration values)
  end for
  $zmat[, p] \leftarrow zvec$ (Save $zvec$ to pth column of $zmat$)
end for
return $[xvec, yvec, zmat]$

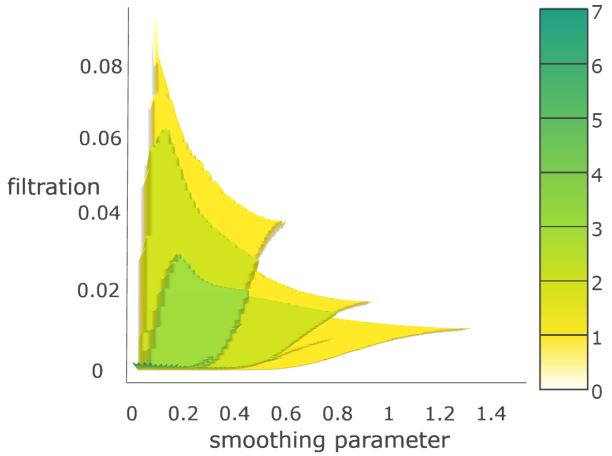


(a) 25 smoothing parameters

(b) 50 smoothing parameters



(c) 75 smoothing parameters



(d) 100 smoothing parameters

Figure 1: Resolution of the persistence terrace according to the number of smoothing parameters. With an increase in the number of smoothing parameters, the resolution increases, although the general picture stays the same.

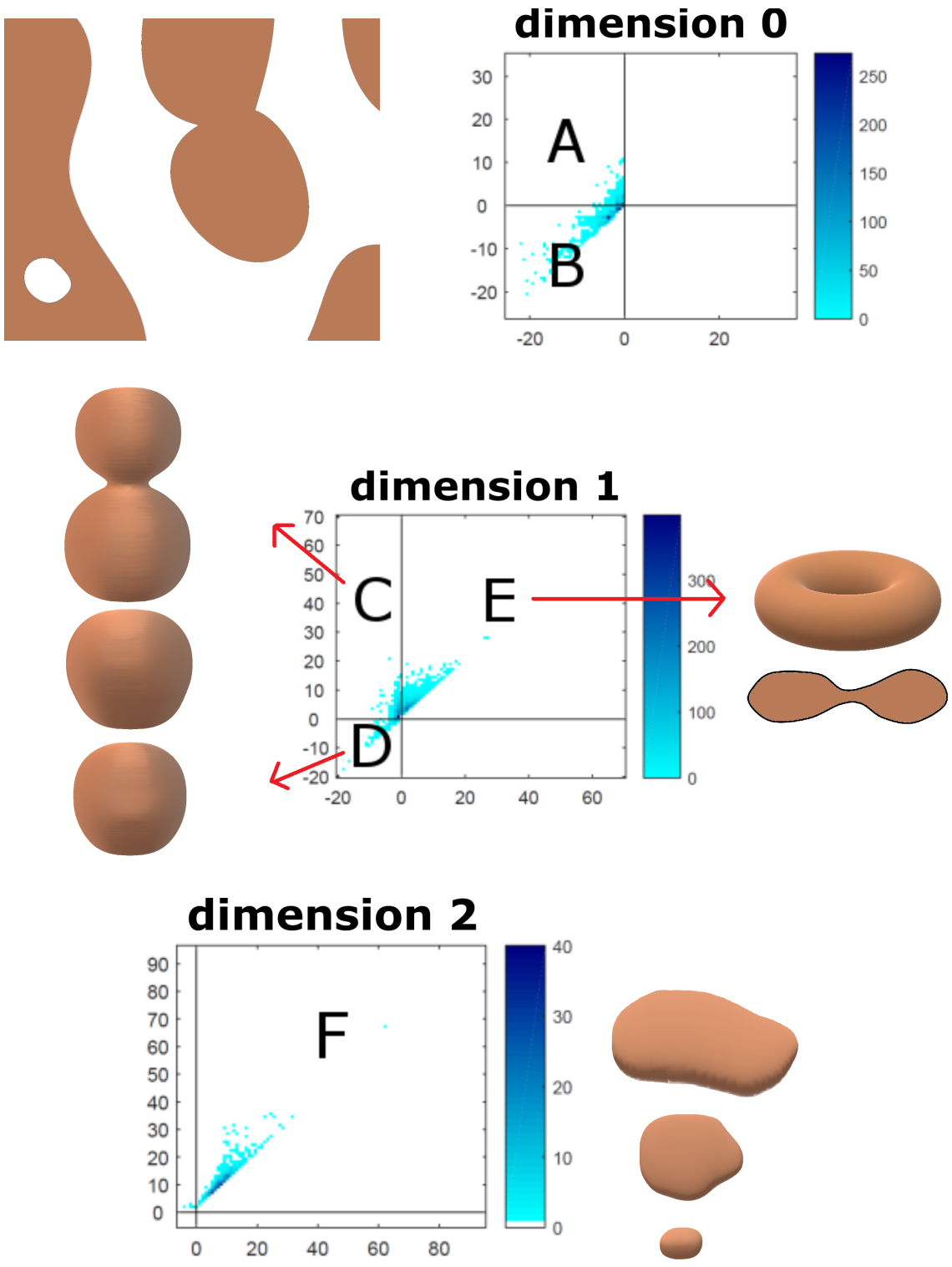
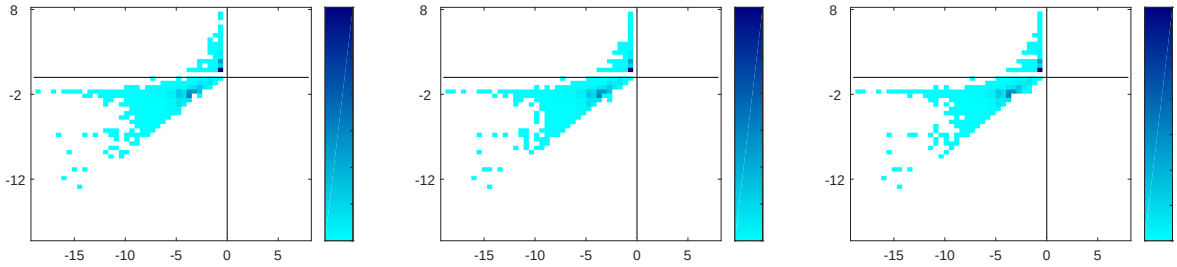


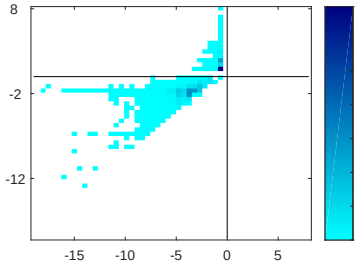
Figure 2: Examples of pore and grain structures in the corresponding dimension zero (top row), one (second row), and two (last row) persistence diagrams.



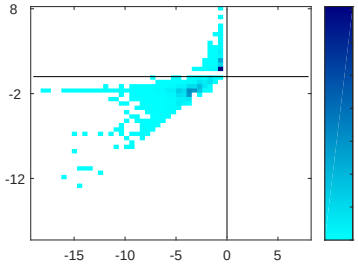
(a) Stress-strain: 2%

(b) Stress-strain: 4%

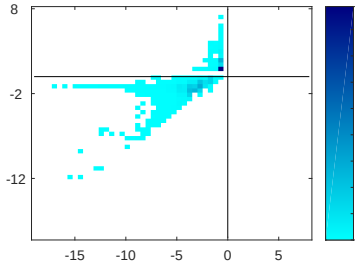
(c) Stress-strain: 6%



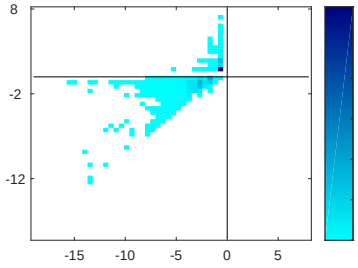
(d) Stress-strain: 8%



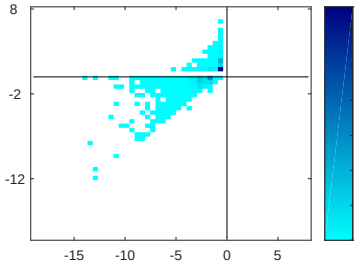
(e) Stress-strain: 10%



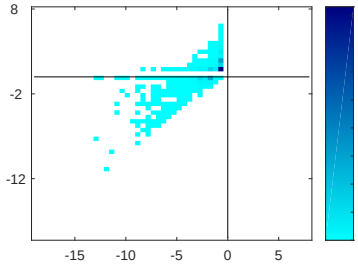
(f) Stress-strain: 20%



(g) Stress-strain: 30%

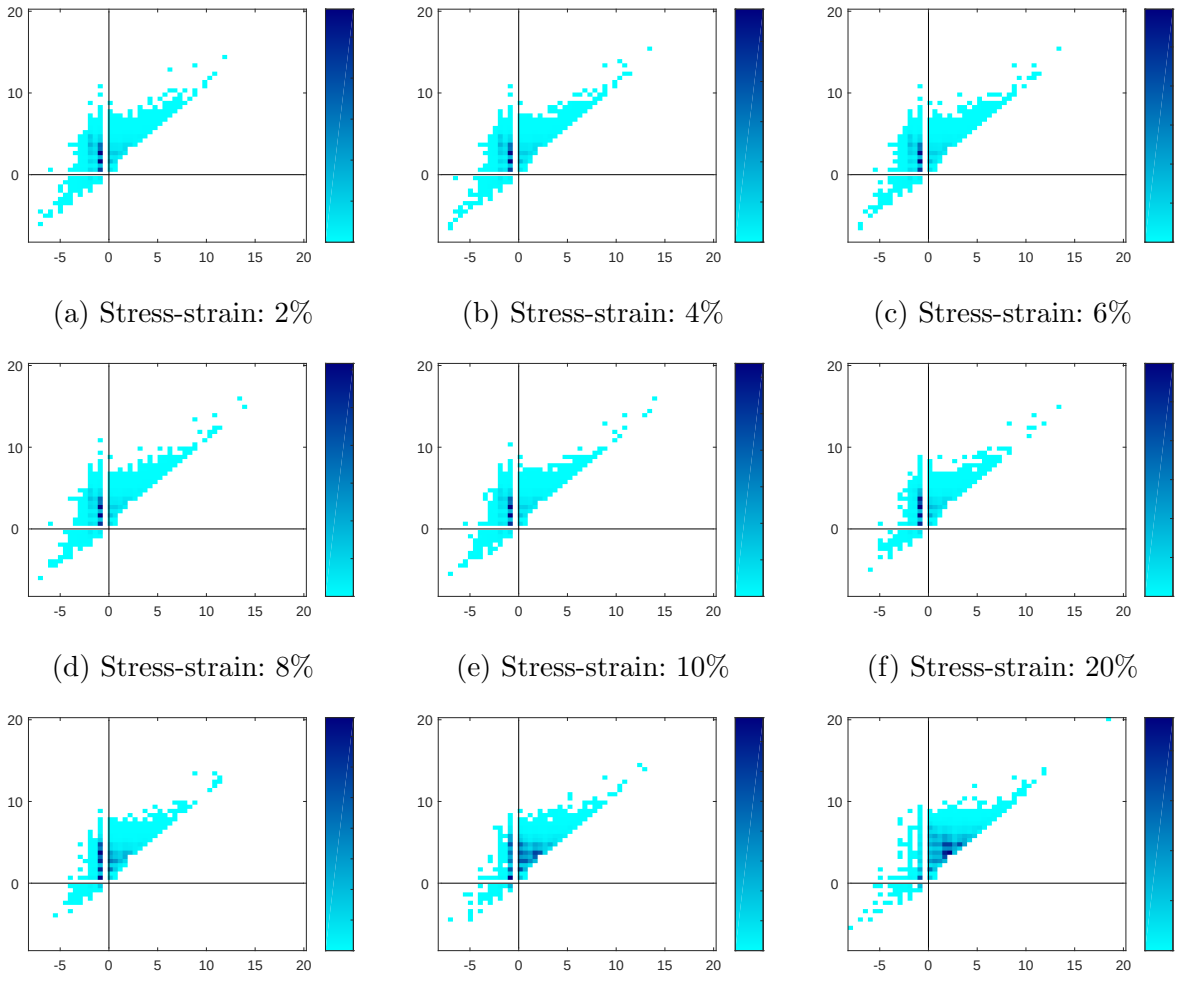


(h) Stress-strain: 40%



(i) Stress-strain: 50%

Figure 3: Dimension 0 persistence diagrams under nine stress-strain levels.



(a) Stress-strain: 2%

(b) Stress-strain: 4%

(c) Stress-strain: 6%

(d) Stress-strain: 8%

(e) Stress-strain: 10%

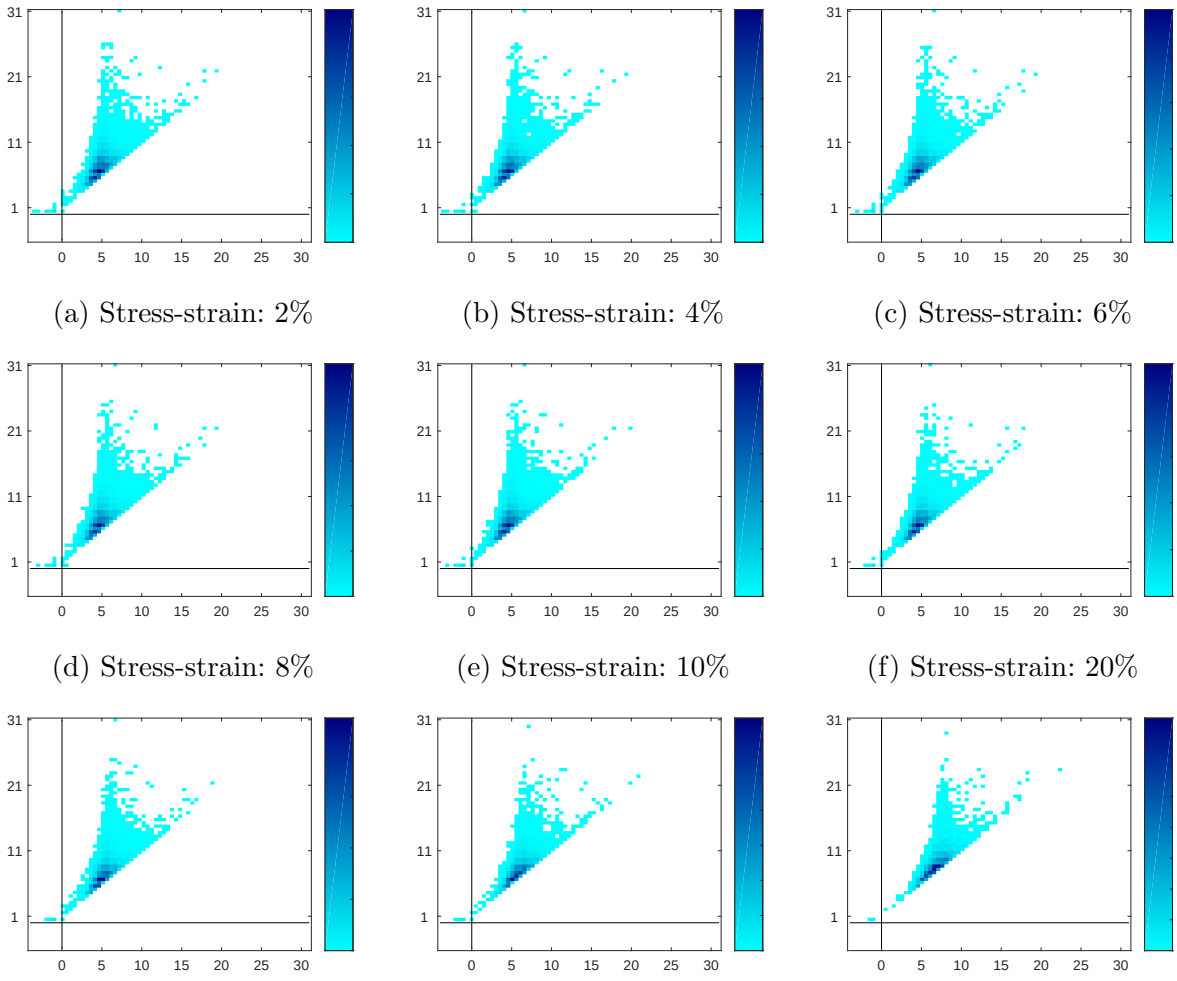
(f) Stress-strain: 20%

(g) Stress-strain: 30%

(h) Stress-strain: 40%

(i) Stress-strain: 50%

Figure 4: Dimension 1 persistence diagrams under nine stress-strain levels.



(a) Stress-strain: 2%

(b) Stress-strain: 4%

(c) Stress-strain: 6%

(d) Stress-strain: 8%

(e) Stress-strain: 10%

(f) Stress-strain: 20%

(g) Stress-strain: 30%

(h) Stress-strain: 40%

(i) Stress-strain: 50%

Figure 5: Dimension 2 persistence diagrams under nine stress-strain levels.

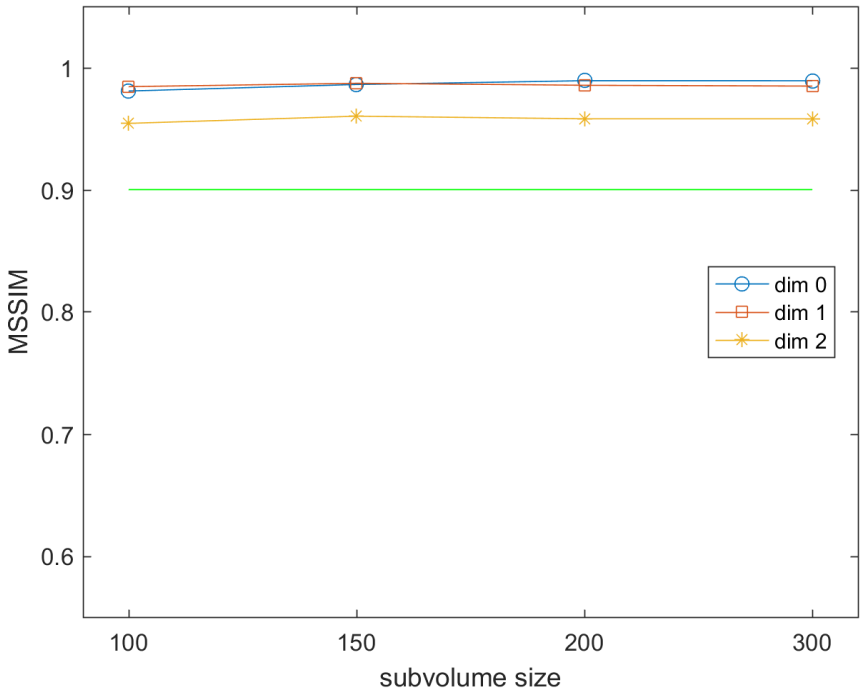


Figure 6: Average SSIM (MSSIM_{PH}) of Rotleigend.