Inaugural-Dissertation

zur Erlangung der Doktorwürde der

Naturwissenschaftlich-Mathematischen
Gesamtfakultät

der

Ruprecht-Karls-Universität
Heidelberg

vorgelegt von
Diplom-Mathematiker

Bastian Alexander Rieck

aus Heidelberg

Tag der mündlichen Prüfung: 28.04.2017

Persistent Homology in Multivariate Data Visualization

by

Bastian Alexander Rieck

Supervisors: Prof. Dr. Heike Leitte
Prof. Dr. Michael Gertz

The fool doth think he is wise,
but the wise man knows himself to be a fool.

—William Shakespeare, As You Like It, Act V, Scene I

Dedicated to the loving memory of my maternal grandmother
Dorothea Barbara Ortlieb (née Beierbach)
1938–2011

Abstract

Technological advances of recent years have changed the way research is done. When describing complex phenomena, it is now possible to measure and model a myriad of different aspects pertaining to them. This increasing number of variables, however, poses significant challenges for the visual analysis and interpretation of such *multivariate data*. Yet, the effective visualization of structures in multivariate data is of paramount importance for building models, forming hypotheses, and understanding intrinsic properties of the underlying phenomena. This thesis provides novel visualization techniques that advance the field of multivariate visual data analysis by helping represent and comprehend the structure of high-dimensional data. In contrast to approaches that focus on visualizing multivariate data directly or by means of their geometrical features, the methods developed in this thesis focus on their topological properties. More precisely, these methods provide structural descriptions that are driven by *persistent homology*, a technique from the emerging field of computational topology.

Such descriptions are developed in two separate parts of this thesis. The first part deals with the *qualitative visualization* of topological features in multivariate data. It presents novel visualization methods that directly depict topological information, thus permitting the comparison of structural features in a qualitative manner. The techniques described in this part serve as low-dimensional representations that make the otherwise high-dimensional topological features accessible. We show how to integrate them into data analysis workflows based on clustering in order to obtain more information about the underlying data. The efficacy of such combined workflows is demonstrated by analysing complex multivariate data sets from cultural heritage and political science, for example, whose structures are hidden to common visualization techniques.

The second part of this thesis is concerned with the *quantitative visualization* of topological features. It describes novel methods that measure different aspects of multivariate data in order to provide quantifiable information about them. Here, the topological characteristics serve as a feature descriptor. Using these descriptors, the visualization techniques in this part focus on augmenting and improving existing data analysis processes. Among others, they deal with the visualization of high-dimensional regression models, the visualization of errors in embeddings of multivariate data, as well as the assessment and visualization of the results of different clustering algorithms.

All the methods presented in this thesis are evaluated and analysed on different data sets in order to show their robustness. This thesis demonstrates that the combination of geometrical and topological methods may support, complement, and surpass existing approaches for multivariate visual data analysis.

Zusammenfassung

Der technologische Fortschritt der letzten Jahre hat die Art, in der Wissenschaft betrieben wird, nachhaltig verändert. Es ist nun möglich, bei der Beschreibung komplexer Phänomene eine Vielzahl von Aspekten zu erfassen. Die immer größer werdende Anzahl an Variablen, die hierzu benötigt werden, stellt existierende Verfahren zur Darstellung *multivariater Daten* vor erhebliche Probleme. Dabei ist gerade die Visualisierung von Strukturen in multivariaten Daten von höchster Wichtigkeit für die Modellierung, das Aufstellen von Hypothesen, sowie das intrinsische Verständnis von Daten. Diese Dissertation stellt neue Visualisierungsmethoden vor, welche die Repräsentation und das Verständnis von Strukturen in hochdimensionalen Daten erlauben. Im Gegensatz zu Methoden, die sich auf die direkte Darstellung von multivariaten Daten beziehen oder deren geometrischen Eigenschaften nutzen, konzentrieren sich die Methoden dieser Dissertation zusätzlich auf die topologischen Eigenschaften von Daten, d.h. auf ihren Zusammenhang. Sie stellen dabei strukturelle Beschreibungen zur Verfügung, die durch das Konzept der *persistenten Homologie* ermöglicht werden.

Derartige Beschreibungen werden in den zwei unterschiedlichen Teilen der vorliegenden Arbeit genutzt. Der erste Teil befasst sich mit *qualitativen Visualisierungen* topologischer Merkmale in Daten. Er stellt neue Visualisierungsmethoden vor, die eine direkte Darstellung topologischer Informationen erlauben und es somit ermöglichen, Strukturen in Daten qualitativ zu vergleichen. Die Methoden in diesem Teil stellen niedrigdimensionale Repräsentationen dar, welche die ansonsten hochdimensionalen topologischen Merkmale erfassbar machen. Wir zeigen, wie sie in Arbeitsabläufe zur Datenanalyse, welche sich bestimmter Clusteringverfahren bedienen, integriert werden können, um eine genauere Beschreibung der zugrundeliegenden Daten zu erhalten. Die Nützlichkeit einer solchen kombinierten Herangehensweise belegt die Arbeit durch die Analyse komplexer multivariater Datensätze, deren Strukturen sich der Visualisierung durch gewöhnliche Methoden entziehen.

Der zweite Teil der Arbeit beschäftigt sich mit der *quantitativen Darstellung* von topologischen Eigenschaften. Er beschreibt neue Methoden, die verschiedene Aspekte von Daten messen, um quantifizierbare Informationen zu erhalten. Die topologischen Charakteristika von Daten dienen somit als Merkmalsbeschreibung. Diese Beschreibungen nutzen wir unter anderem zur Darstellung von hochdimensionalen Modellen in der Regressionsanalyse, zur Visualisierung von fehlerhaften Regionen in Einbettungen multivariater Daten, sowie zur Bewertung und Darstellung von Ergebnissen unterschiedlicher Clusteringverfahren. Alle vorgestellten Methoden werden zudem auf ihre Robustheit hin untersucht, indem sie auf unterschiedlichen Datensätze evaluiert werden. Diese Arbeit zeigt damit, dass die Kombination von geometrischen und topologischen Methoden bereits bekannte Ansätze zur visuellen multivariaten Datenanalyse unterstützen, ergänzen, und sogar übertreffen kann.

Acknowledgements

I owe my deepest gratitude to my supervisor Prof. Dr. Heike Leitte. She gave me guidance and support when and where I needed it the most, but was always willing to give me leave to find my own paths. Her enthusiasm and kind words spurred me on to developing my own style in writing, researching, teaching, and mentoring. These are truly the best gifts anyone could have received. Furthermore, I would like to extend my thanks to my second supervisor, Prof. Dr. Michael Gertz, with whom I enjoyed fruitful discussions that opened my eyes beyond the field of visualization. I am also grateful to Prof. Dr. Filip Sadlo and P. D. Dr. Wolfgang Merkle for being part of my thesis committee.

Over the years, I have had the pleasure to discuss my work with many people. No amount of stale words can do their influence justice. However, Marcus Aurelius—my mentor in many things—taught me to attempt it anyway, and so I shall. From Andreas Beyer, I always obtained the right nudge at the right time. From Daniel Beyer, I learned the value of endurance and how to question my work without rancour. From Bartosz Bogacz, the merits of a more realistic perspective. From Lutz Büch, calmness and serenity. From Hamish Carr, the desire to look for the intuition behind concepts. From Christoph Garth, a healthy mixture of enthusiasm and pragmatism. From Katja Hauser, the importance of resting body and mind. From Christian Heine, the drive to search for knowledge beyond my own area of work. From Markus Kurz, exuberant joy and how to love what I do. From Matthias Maier, to put more trust in my own work. From Julia Portl, the ambition to let graphics tell a story. From Maria Rupprecht, the faith in things both permanent and impermanent. From Filip Sadlo, the awe, humility, and bearing that is the hallmark of a scientist. From Julia Seifert, an unwavering positivity and the belief in myself. From Julien Tierny, an appetite for self-contained explanations.

I am fortunate to have experienced the influence of so many remarkable people in my life. They contributed to shaping me into the person I am today. I am indebted to my teachers, in particular Markus Banagl, Sigrid Böge, Susanne Krömker, and Matthias Kreck, who instilled me with a passion for algebra and topology. Furthermore, I want to thank my students Alexander Eck, Daniel Beyer, Jan Greulich, Markus Kurz, Karsten Hanser, Katja Hauser, and Sophia Stahl, for giving me the honour of advising them. It was a privilege.

I gratefully acknowledge the financial support for travelling and the stipend I received for parts of my doctoral research by the *Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences* (HGS MathComp), represented by Dr. Michael J. Winckler.

I am also thankful for the people who read this thesis completely or in parts. Even though the number of pages seemed daunting, they did not despair and helped me accomplish my work. Great thanks are due to Daniel 'The Proofreading Machine' Beyer, Katja Hauser, Markus Kurz, Florian Rieck, Filip Sadlo, Maria Rupprecht, and Niky Yaneva.

I wish to thank my friends and colleagues Andreas Beyer, Bartosz Bogacz, Alexander Eck, Jens Fangerau, Christopher Kappe, Ole Klein, Susanne Krömker, Hubert Mara, Julia Portl, Filip Sadlo, and Boyan Zheng, and all the many marvellous companions on this journey. You provided a friendly atmosphere for which I am grateful. Moreover, I thank the administrative staff of HGS MathComp, IWR, and TU Kaiserslautern, who always tamed the bureaucratic beast and removed the frictions that inevitably arise when it comes to IT, funding, extensions, or travel expenses. I was also fortunate to be a part of two research groups in two different locations. Even though my visits to Kaiserslautern were brief, I was always received with open arms. I am grateful that Prof. Dr. Hans Hagen takes care to maintain such a welcoming environment and is willing to go to any lengths to support his students. Last, but certainly not least, I thank my parents, my brother, and the rest of my family for their continuous support. Whatever I did to deserve you, it could not have been enough.

Thank you.

I would not wish
Any companion in the world but you,
Nor can imagination form a shape,
Besides yourself, to like of.

—William Shakespeare, The Tempest, Act III, Scene I

Contents

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 What is topology? . . . 2
- 1.3 Why topology? . . . 4
- 1.4 Aims & scope of this thesis . . . 5
- 1.5 Contributions . . . 7
- 1.6 Structure of this thesis . . . 9

2 Related work 13
- 2.1 Multivariate statistics . . . 13
- 2.2 Multivariate visualizations . . . 14
- 2.3 Glyph-based visualizations . . . 16
- 2.4 Dimensionality reduction methods . . . 17
- 2.5 Projection-based visualizations . . . 18
- 2.6 Morse theory . . . 18
- 2.7 Feature-based & hybrid methods . . . 22

3 Algebraic topology 25
- 3.1 Topological spaces & their invariants . . . 26
- 3.2 Simplicial homology . . . 27
- 3.3 Relative simplicial homology . . . 34
- 3.4 Calculating simplicial homology . . . 35
- 3.5 Discussion . . . 38

4 Persistent homology 41
- 4.1 Nerves, covers, and complexes . . . 42
- 4.2 Calculating the Vietoris–Rips complex . . . 47
- 4.3 Calculating 0-dimensional persistent homology . . . 51
- 4.4 Calculating persistent homology . . . 55
- 4.5 Visualizing persistent homology . . . 64
  - 4.5.1 Persistence diagrams . . . 65

4.5.2 Persistence barcodes . . . 67
4.6 Quantifying topological similarity . . . 68
4.6.1 Distances between persistence diagrams . . . 68
4.6.2 Stability . . . 72
4.6.3 Comparing topological & geometrical distances . . . 75
4.7 Discussion . . . 77

I Visualizing qualitative topological information 81

5 Topological fingerprints in cluster analysis 83
5.1 Persistence rings . . . 83
5.2 Combining topological analysis & clustering . . . 87
5.3 Persistence-based clustering . . . 90
5.3.1 Density estimation . . . 91
5.3.2 Peak estimation using persistent homology . . . 92
5.3.3 An example . . . 94
5.4 Rips graph parameter selection . . . 96
5.5 Application to synthetic test data . . . 98
5.6 Application to cultural heritage data . . . 100
5.6.1 Multi-scale integral invariant filters . . . 102
5.6.2 Synthetic data set . . . 106
5.6.3 Real data set . . . 108
5.7 Discussion . . . 111

6 Structural analysis of point clouds using simplicial chains 115
6.1 Why do we need geometrical information? . . . 116
6.1.1 The localization problem . . . 117
6.1.2 A notion of conciseness . . . 117
6.2 Localizing simplicial chains . . . 118
6.2.1 Approximating & extending geodesic distances . . . 123
6.2.2 Finding the smallest geodesic ball . . . 123
6.2.3 Removing a homology class from $\mathcal{V}_\epsilon$ . . . 126
6.3 The simplicial chain graph . . . 126
6.3.1 Properties . . . 130
6.3.2 Stability & extensions . . . 131
6.4 Analysis of several data sets . . . 132
6.4.1 Voting data . . . 132

6.4.2 Tropical Atmosphere Ocean array data . . . 137
6.5 Discussion . . . 142

II Visualizing quantitative topological information 145

7 Evaluating embeddings 147
7.1 Dimensionality reduction methods . . . 148
7.2 Quality measures . . . 151
7.3 Agreement analysis . . . 156
7.4 Results of agreement analysis . . . 159
7.4.1 Handwritten digits . . . 160
7.4.2 Concrete compressive strength . . . 166
7.5 Data descriptors . . . 170
7.6 Using data descriptors to evaluate embeddings . . . 172
7.7 An example . . . 176
7.7.1 Global quality . . . 177
7.7.2 Local quality . . . 179
7.8 Stability & performance . . . 180
7.9 Results . . . 184
7.9.1 Synthetic faces . . . 184
7.9.2 Concrete compressive strength . . . 186
7.9.3 Climate data . . . 189
7.10 Discussion . . . 191

8 Landscape metaphors for multivariate data 197
8.1 Visualizing regression analysis models . . . 197
8.1.1 Related work . . . 198
8.1.2 Quality measures for regression analysis . . . 199
8.1.3 A quality measure based on persistent homology . . . 201
8.1.4 Solubility analysis . . . 204
8.2 Visualizing properties of embeddings . . . 214
8.2.1 Depicting multiple data descriptors . . . 214
8.2.2 Results . . . 216
8.3 Discussion . . . 221

9 Assessing & visualizing clusterings 225
9.1 Related work . . . 226

9.2 Methods . . . 228
9.2.1 Choosing a data descriptor . . . 228
9.2.2 Extended persistent homology . . . 229
9.2.3 Total persistence . . . 232
9.2.4 Assessing clusterings . . . 233
9.2.5 Comparison with existing clustering validity indices . . . 238
9.2.6 Visualization methods . . . 244
9.3 Results . . . 248
9.3.1 'Iris flower' data . . . 249
9.3.2 'Olive oils' data . . . 253
9.3.3 'El Niño' data . . . 257
9.4 Discussion . . . 260

10 Conclusion 263

Acronyms 267

Glossary 269

Index 271

Bibliography 275

1 Introduction

The last few decades demonstrated that we live in an age that is teeming with data. It is not only the ever-increasing amount of internet users—more than 3 billion in 2015 according to the *International Telecommunication Union* of the United Nations—that results in more and more data being created. The sciences also regularly generate large data sets along with their experiments. It has been said that nowadays, data sets have the same significance as the microscope for scientists in the 18th century[1]. With more and more data sets being available to the public for scientific and non-scientific purposes[2], significant challenges arise—the most pressing being *how to make sense* of these data. Two independent factors make this endeavour difficult. First, the sheer *amount* of data requires new storage strategies and high-performance algorithms. Second, the *dimensionality* of a data set—measured, for example, by the number of variables—necessitates novel ways for processing, internalizing, and visualizing it for humans. In this thesis, we only focus on the second aspect and develop methods that are capable of visualizing high-dimensional data.

1.1 Motivation

While there is some controversy whether the figural deluge of high-dimensional data sets resulted in the creation of a new field of science—*data science*—or not [131], scientists agree on one thing: The overabundance of data requires new instruments for looking at them. The goal of this thesis is to provide and describe some of these new instruments, which appear in the form of methods from computational topology.

Developing and deploying new instruments has a long-standing tradition in the visualization community. John Tukey, for example, is seen by many as the forefather of modern data analysis. He proposed interactive visualizations for making sense of multivariate data. As early as 1962, Tukey [370] thus envisioned the potential of analysing data in an inferential

[1] Erik Brynjolfsson & Andrew McAfee, 'The big data boom is the innovation story of our time'. *The Atlantic*, November 2011.

[2] The platform for the U.S. Government's open data, data.gov, contains almost 200,000 data sets at the last count in early 2016. This does not even include various other initiatives by U.S. cities, for example Seattle, who provide their own platforms for accessing data.

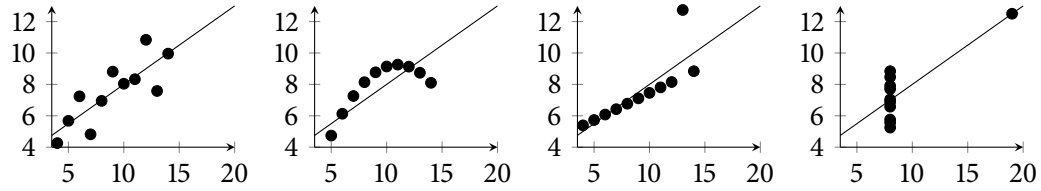


Figure 1.1: Anscombe's quartet. The mean and sample variance of x and y is equal in each data set. Likewise, Pearson's correlation coefficient is 0.816 and the linear regression line is given by $y = 3 + 1/2x$.

and incisive[3] manner. Several years later, he coined the term *exploratory data analysis* (EDA) for these procedures. In an influential monograph of the same title [369], Tukey argued that analysing data without preconceived notions and models helps in forming hypotheses, which in turn lead to models that accurately describe the properties of data. This view is shared by scientists within the visualization community who know that summary statistics tend to fail when it comes to capturing interesting patterns in data. The classical example is *Anscombe's quartet*, a collection of data sets sharing the same summary statistics. However, when shown as a scatterplot, every data set exhibits unique properties. Figure 1.1 demonstrates this.

✦

This thesis presents and develops novel methods for the visualization of multivariate data. All these methods are firmly rooted in algebraic topology, the branch of mathematics that deals with measuring connectivity of spaces. More specifically, we shall employ concepts from the newly-emerging area of *topological data analysis* (TDA). The subsequent sections give a brief overview of algebraic topology and motivate its use. We pay particular attention to highlighting the advantages of topological methods over traditional data analysis methods, but also comment on their shortcomings.

1.2 What is topology?

Topology studies certain mathematical objects, the *topological spaces*, using concepts such as transformations and invariants. The ideas of modern topology can be traced back to Gottfried Wilhelm Leibniz, who referred to this field as *geometria situs*[4] and *analysis situs*[5],

[3] Tukey understood that some patterns in data cannot be perceived by 'simple and direct examination of the raw data'. The obstructions thus need to be *cut* away, leading to the term 'incisive'.

[4] Literally the 'geometry of (a) place'. This indicates that (at least initially) geometry played an important role in describing connectivity.

[5] Literally the 'picking apart of (a) place'. This term indicates that a space is to be closely examined to expose its core properties.

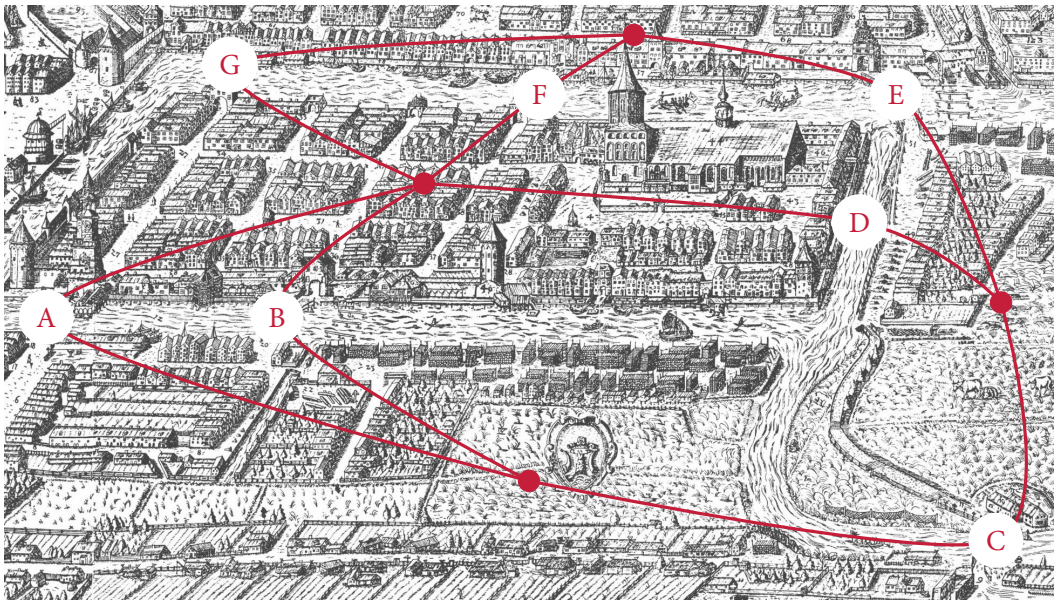
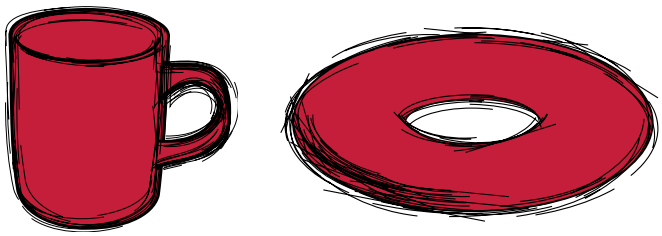


Figure 1.2: The seven bridges of Königsberg. Leonhard Euler was tasked to find out whether it was possible to devise a walk through the city that only used every bridge exactly once. By reducing the problem to a graph, Euler was able to show that such a solution does not exist because the desired walk would require that the *degree* of every vertex in the graph is even.

and Leonhard Euler, whose 'Seven bridges of Königsberg' problem constitutes one of the first theorems in the field. Figure 1.2 depicts a slightly-abstracted version of the problem. The question was whether it was possible to walk through the city of Königsberg, crossing every bridge exactly once, and returning to the starting position. In 1736, Leonhard Euler approached this problem and realized that the geometrical information is irrelevant here. He thus reduced the map to a simple graph in which a vertex represents a land mass and an edge indicates that two land masses are connected by a bridge. This course of action is characteristic for topological methods. Everything that does not contribute to the connectivity of the problem is ignored. Euler realized that in order for the desired walk to exist, every vertex in the graph needs to have an even degree. This follows from the fact that as one enters a land mass via one bridge, one needs to exit it via another one. However, the graph only has vertices with an odd degree. Thus, the desired walk does not exist.

✦

At its core, modern topology analyses topological spaces, such as subsets of some $\mathbb{R}^n$ with a notion of distance. Beginning with this simple definition, a variety of objects with different properties can be defined. *Manifolds*, a special class of such objects, play a central role in this setting. Informally (we shall encounter more formal definitions later on), a manifold 'looks and



A coffee mug A torus

Figure 1.3: A coffee mug and a torus. In the sense of algebraic topology, these two objects are the same because they can be transformed into each other without tearing anything apart.

behaves like some $\mathbb{R}^n$', meaning that it exhibits a smooth local structure. One goal of topology involves the classification of manifolds so that one may decide whether two descriptions of a manifold actually refer to the same manifold. For low-dimensional manifolds, such as the torus, this is rather easy; see Figure 1.3 for an example. In general, classifying manifolds up to homeomorphism turns out to be infeasible. A less discriminative but computationally feasible classification requires some algebraic machinery, known as homology groups. These groups count the holes in a manifold. The holes of a torus, for instance, are different from the holes of a hollow sphere. Hence, these two manifolds are fundamentally different. Again, we will re-encounter this example later on. About a decade ago, Edelsbrunner and Harer [142] discovered that methods from algebraic topology could be of use when analysing real-world data sets. In their seminal paper, Edelsbrunner et al. [148] assume that a given discrete, noisy data set has actually been sampled from some unknown high-dimensional manifold. This 'manifold assumption' is known in many other fields that try to analyse data sets, albeit under different names [31, 133]. In image analysis, for example, it is very common to assume that the discrete samples one works with are actually part of a continuous object in a high-dimensional space. Having seen that this assumption is justified, Edelsbrunner et al. describe their vision of a homology theory for discretely-sampled data. This marked the beginning of topological data analysis based on methods from algebraic topology.

1.3 Why topology?

What is the appeal of using topology-based methods, especially given the strong performance by geometrical and statistical methods over the last decades? *Persistent homology*, the main concept used in this thesis, has numerous beneficial properties for data analysis:

- It is independent of coordinate definitions and coordinate frames because it only depends on pairwise distances in the data. The same patterns can thus be found regardless of the orientation of data, for example.

- It is invariant under many deformations—as long as the underlying space is not 'torn apart', its topology will not change.
- It automatically assigns features in a data set information about their scale, which accommodates the fact that real-world data commonly contain patterns at not just a *single* scale but at *multiple* scales.

Furthermore, numerous stability theorems [92, 104, 105] are known for the constructions used in this thesis—we shall describe them in more detail in Chapter 4. As a consequence, the behaviour of persistent homology under the assumption of noise—an inevitability when dealing with real-world data—is well-studied and well-known, which makes persistent homology very suitable for data analysis. However, as the famous adage 'There ain't no such thing as a free lunch' reminds us, the stability, robustness, and expressiveness of persistent homology come with a price. First, and most crucially, there is the abstractness of features it calculates. The visualization techniques presented in this thesis mitigate this issue by showing more accessible representations of topological information. Another issue concerns the complexity requirements of calculating higher-dimensional topological features. None of the known approaches for persistent homology exhibits particularly good scalability properties. While some progress [220] has been made so that at least the calculation of distances between persistence diagrams may be improved, the computational requirements are overall very high. This leaves a large number of open topics for future research, which we discuss in the individual chapters and return to in the concluding chapter.

1.4 Aims & scope of this thesis

This thesis is motivated by a simple question: How can methods from algebraic topology support the visualization—and subsequent analysis—of scientific data? Research in topological data analysis has so far been performed with a very mathematical perspective. The goal of this thesis is to make the wealth of topological methods accessible and usable in other domains. This requires effective visualization techniques that express patterns in high-dimensional space. In the context of visualizing and comprehending *multivariate data*, we will adopt a pragmatic view and consider a multivariate data set to consist of a set of points from a d-dimensional Euclidean space $\mathbb{R}^d$, possibly equipped with some notion of distance measure. We have the following definition.

Definition 1.1 (Multivariate data set). A multivariate data set consists of measurements or observations of two or more variables that are not necessarily independent. In particular, the measurements are *unstructured* in the sense that there is no underlying order such as a grid or graph. The number of variables in each instance must not change, however.

A variable in this sense is something that can be easily quantified, such as *length* or *age*, or requires more advanced measurement techniques, such as determining a *protein docking site* or a *relationship* in a social network—hence, our definition is not restricted to certain data sources. We will also refer to multivariate data as *point cloud data* to signify that they exist in no particular arrangement. The definition is sufficiently broad to encompass all sorts of interesting data sets, such as oceanographic measurements, shape descriptors of handwritten digits, and feature vectors of cultural heritage data. Within this context, the term 'high-dimensional' refers to the number of variables that is usually much larger than three. Mathematically speaking, this only pertains to the *ambient dimension* of the data—the *intrinsic dimension*, i.e. the amount of variables that are actually required to describe the data succinctly, is often much smaller. We shall investigate several examples of this later on, especially in Chapter 7, where we develop a new method for assessing the suitability of dimensionality reduction methods.



Given a set of measurements with different variables, we follow the conceptual model of Stevens [350], who proposed a classification for them. According to this classification, a variable may have a nominal, ordinal, interval, or ratio scale. *Nominal* scales, the most primitive ones, permit only to determine whether two values are equal or not—one may think of names of persons, for example. *Ordinal* scales additionally permit the ranking and sorting of values but nothing else. Clothing sizes, for example, may be ordered from extra-small to extra-large, but we cannot define what a 'large minus medium' shirt is going to look like. *Interval* scales furthermore permit the determination of equality of intervals and differences, i.e. there is an underlying metric. The zero point of such a scale may not be well-defined, though. A typical example of an interval scale is room temperature measured in degrees Celsius. It is perfectly valid to calculate the difference between two room temperatures in order to determine whether a room has been heated up or cooled down. However, it is not justified to observe that a room is 'twice as hot' as another room. For these statements to make sense, we require a *ratio* scale. These scales permit determining the equality of ratios of values. Room temperature measured in degrees Kelvin, for example, is an absolute scale.

Keeping the theory of measurements in mind, we thus permit Definition 1.1 to apply to data sets whose variables have different measurement scales. We may of course have other information available. For example, there may be specific attributes, such as a physical location, or variables may be ordered in a grid. However, our methods explicitly do not require more information than the variables and the observations themselves, making them almost universally applicable.

1.5 Contributions

The major contributions of this thesis are:

- In Chapter 5, we develop a novel visualization method for topological information—the *persistence ring*—which has several perceptual advantages over existing approaches. As there is no fixed order in which to display topological attributes, we devise a layout heuristic that maximizes the amount of discriminative information that can be displayed without overlaps. Furthermore, we establish a new workflow for analysing complex multivariate data sets, in which we integrate the persistence ring visualization and show its efficacy [318].

- In Chapter 6, we present a new algorithm for integrating geometrical information into the calculation of persistent homology. This combination of geometry and topology increases the discriminative properties of our technique and permits it to detect a larger class of features. Our algorithm uses an optimization strategy to provide topological features that are as concise as possible with respect to their geometry. We use the algorithm as the foundation for a novel visualization technique—the *simplicial chain graph* of structural information in multivariate point clouds [316].

- In Chapter 7, we detail a novel method for assessing embeddings of high-dimensional data sets. We first develop a new algorithm for the decomposition of scalar fields. This lets us derive a way of measuring the *agreement* of several quality measures for high-dimensional embeddings. Following this, we establish a new way of calculating persistent homology that permits us to evaluate embeddings both globally and locally. We derive an upper bound for the resulting values, which makes it possible to declare certain embeddings to be unsuitable because they are incapable of preserving geometrical–topological information. Our evaluation is based on a novel set of special functions, the data descriptors, that we use to quantify certain salient properties of multivariate data sets. [310, 315]

- In Chapter 8, we present *model landscapes* and *data descriptor landscapes*, two topology-driven embeddings for depicting structural dissimilarities of multivariate data sets in a manner that may be easily understood. We show that these landscape metaphors permit the quick assessment of different high-dimensional data sets. Moreover, we show that persistent homology does not suffer from the same instabilities and limitations as existing quality measures. [311, 313]

- In Chapter 9, we develop a new topology-based quality measure that permits assessing high-dimensional clusterings both globally and locally without requiring class labels.

> We then go on to integrate this quality measure into two new visualizations, the *clustering similarity graph* and the *cluster map*, which permit a holistic analysis of a clustering. Furthermore, we use numerous example data sets to show that our topology-based measure outperforms existing quality measure, both in terms of stability & robustness and in expressive power. [314]

This thesis is partially based on several publications by the author, which are indicated using brackets in the list above. In comparison to the original publications, this thesis contains novel experiments and a more in-depth treatment of results. A detailed list of publications in reverse chronological order follows.

- B. Rieck and H. Leitte. 'Exploring and comparing clusterings of multivariate data sets using persistent homology'. *Computer Graphics Forum* 35:3, 2016, pp. 81–90. DOI: 10.1111/cgf.12884
- B. Rieck and H. Leitte. 'Comparing dimensionality reduction methods using data descriptor landscapes'. In: *Symposium on Visualization in Data Science* (VDS) *at IEEE VIS*. Chicago, IL, USA, 2015
- B. Rieck and H. Leitte. 'Persistent homology for the evaluation of dimensionality reduction schemes'. *Computer Graphics Forum* 34:3, 2015, pp. 431–440. DOI: 10.1111/cgf.12655
- B. Rieck and H. Leitte. 'Agreement analysis of quality measures for dimensionality reduction'. In: *Workshop on Topology-Based Methods in Visualization* (TopoInVis). To appear in *Topological Methods in Data Analysis and Visualization IV*. Annweiler, Germany, 2015
- B. Rieck and H. Leitte. 'Structural analysis of multivariate point clouds using simplicial chains'. *Computer Graphics Forum* 33:8, 2014, pp. 28–37. DOI: 10.1111/cgf.12398
- B. Rieck and H. Leitte. 'Enhancing comparative model analysis using persistent homology'. In: *Workshop on Visualization for Predictive Analytics at IEEE VIS*. Paris, France, 2014
- B. Rieck, H. Mara and H. Leitte. 'Multivariate data analysis using persistence-based filtering and topological signatures'. *IEEE Transactions on Visualization and Computer Graphics* 18:12, 2012, pp. 2382–2391. DOI: 10.1109/TVCG.2012.248

The author also contributed to the following publications that do not directly relate to the core topic of this thesis:

- B. Rieck and H. Leitte. '*Shall I compare thee to a network?* — Visualizing the topological structure of Shakespeare's plays'. In: *Workshop on Visualization for the Digital Humanities at IEEE VIS*. Baltimore, MD, USA, 2016
- J. Fangerau, B. Höckendorf, B. Rieck, C. Heine, J. Wittbrodt and H. Leitte. 'Interactive similarity analysis and error detection in large tree collections'. In: *Visualization in Medicine and Life Sciences III. Towards making an impact*. Ed. by L. Linsen, B. Hamann and H.-C. Hege. Mathematics and Visualization. Springer, Cham, Switzerland, 2016, pp. 287–307. DOI: 10.1007/978-3-319-24523-2_13
- B. Rieck, H. Mara and S. Krömker. 'Unwrapping highly-detailed 3D meshes of rotationally symmetric man-made objects'. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* II-5/W1, 2013, pp. 259–264. DOI: 10.5194/isprsannals-II-5-W1-259-2013

1.6 Structure of this thesis

We shall start with a discussion of related work in Chapter 2. This chapter provides an overview of methods for multivariate data visualization, adopting viewpoints from statistics, visualization, and dimensionality reduction. Furthermore, it briefly introduces related topological methods and concepts, such as *Morse theory*, which are intrinsically connected to computational topology in general.

Chapter 3 explains important concepts from *algebraic topology*. This is required in order to obtain a firm understanding of persistent homology later on. Choosing this approach also makes this thesis self-contained, so that readers without a strong background in topology will be able to benefit from its contents.

Chapter 4 describes how to calculate *persistent homology* for real-world data sets. It also contains important correctness proofs and discusses two standard visualizations of topological features—*persistence diagrams* and *persistence barcodes*. This discussion is followed by an introduction of different notions of distances between topological feature descriptors. After describing algorithms for their computation and analysing their stability properties, the chapter compares topological distances to function space distances, which are more commonly used. The comparison demonstrates the benefits of topology-based distances, especially in the presence of noise.

✦

Thus, the introductory part of the thesis is concluded. The first part pertaining to the analysis of real-world data sets deals with the visualization of qualitative topological information.

Chapter 5 first presents *persistence rings*, a novel visualization technique for topological attributes in high-dimensional data sets. This technique requires the development of an optimized layout strategy to ensure that all available topological information is displayed concisely and without overlaps. Chapter 5 also discusses the integration of topological information into a standard clustering analysis workflow and describes a clustering algorithm that combines peak estimation with concepts from persistent homology. Moreover, the chapter analyses the stability of the proposed workflow and demonstrates its efficacy on complicated real-world data sets that are not amenable to standard multivariate data analysis techniques.

Chapter 6 takes a different viewpoint and presents a novel visualization—the *simplicial chain graph*—of the connectivity of a high-dimensional point cloud. The *simplicial chain graph* is based on the amalgamation of geometrical and topological features. This endeavour also requires the description of a novel algorithm for the *localization* of topological features, which Chapter 6 provides, describes, and analyses in great detail. Finally, the merits of the *simplicial chain graph* are discussed by means of high-dimensional 'multi-run' data sets, comprising complex varying behaviour. This concludes the first part of this thesis.

The second part deals with the visualization and analysis of quantitative topological information, in effect treating persistent homology as a *feature extraction* algorithm. This requires the usage of the previously-introduced metrics between topological feature descriptors.

Having already established the practical utility of using persistent homology as a feature descriptor, Chapter 7 presents a generic workflow—including novel visualizations and topology-based quality measures—for the quantitative analysis of multivariate data sets by means of persistent homology. This is demonstrated on a highly-relevant topic for data analysis, namely the evaluation of different dimensionality reduction methods. This evaluation has two parts. First, the *agreement* of different quality measures is being investigated, using a new scalar field decomposition algorithm. This leads to an easy-to-understand visualization in which parts of the data that feature a different error distribution than the rest are highlighted. Second, after introducing a novel set of feature descriptors for high-dimensional data sets—the *data descriptors*—the chapter presents a generic workflow for the comparison and analysis of embeddings. The efficacy of this method is demonstrated on synthetic data sets—in order to have a ground truth— as well as on complex multivariate data sets from real-world applications.

Following this, Chapter 8 presents a novel visualization, based on the *landscape metaphor*, in order to visualize complex multivariate data sets under multiple aspects. The chapter contains a detailed quality analysis of high-dimensional regression models, and the novel

topological approach is shown to surpass common quality measures. As a second use case, the chapter analyses numerous embeddings of complex high-dimensional data sets with respect to their 'topological quality'.

Chapter 9 assesses and visualizes clusterings based on their topological attributes. The chapter introduces novel global and local quality measures for clusters and clusterings, followed by an in-depth comparison to existing measures to assess clusterings. It turns out that in the absence of label information, the new measure outperforms the previous measures. Chapter 9 also provides two novel visualizations, the *clustering similarity graph*, which is capable of comparing multiple clusterings with each other, and the *cluster map*, which shows clusterings on a local level. The chapter demonstrates the utility of these visualizations, coupled with our novel measures, by analysing multiple data sets with varying complexities. The thesis ends with Chapter 10, which briefly summarizes the contributions, discusses them, and points out new areas of research that arise as a result of this work.

2 Related work

This chapter briefly reviews existing literature on the visualization of multivariate data and brings the methods presented here into context. In addition, it pinpoints where the methods in this thesis may support, complement, or even surpass existing methods for analysing multivariate data. Moreover, to acquaint the reader with the field of visualization, this chapter also presents examples of several common visualization techniques that will be used throughout the thesis.

2.1 Multivariate statistics

We first shortly explain the statistical perspective on multivariate data. Mathematical statistics presumes little to no knowledge about a data set to draw some inferences from it. Given a set of multivariate observations, the classical construction is to calculate a *sample mean* and a *sample covariance matrix*. If data are assumed to follow a multivariate normal distribution, several inferences about a sample mean vector may be made. Chiefly among those is Hotelling's T^2 test [203], which permits hypothesis testing in the multivariate setting, i.e. the ability to determine whether two multivariate sets of observations have been drawn from the same underlying distribution. Similar calculations may be made for hypothesis testing of covariance matrices, for example. We refer to the classical textbooks by Mardia et al. [261] or Anderson [7] for more details.

While this view is extremely precise, it lacks expressive power. When analysing multivariate data, we require more information about their structure. As a consequence, many publications are concerned with multivariate data analysis in the context of data models. The basic idea is to assume that some of the variables are *predictor* variables and one or more variables are *response* variables. A classical example is the *linear multivariate regression* model

$$\mathbf{Y} = \mathbf{X} \cdot \mathbf{B} + \mathbf{U}, \tag{2.1}$$

where $\mathbf{Y}$ is an $n \times p$ matrix of p response variables, $\mathbf{X}$ is an $n \times q$ matrix of q predictor variables, $\mathbf{B}$ is a $q \times q$ matrix of unknown regression parameters, and $\mathbf{U}$ is an $n \times r$ matrix of noise. The linear regression model must be robust in the sense that perturbations and outliers do not

affect the matrix **B** too much. See Rousseeuw and Leroy [321] for an extensive introduction into the topic of robust regression methods.

The model-based approach is very useful but it is often not applicable for real-world data sets because the assumptions that need to be made are unclear. In the opinion of the author, visualizations should eventually lead to a model so that an analyst may hypothesize about the data. Following this line of thought, another large branch of mathematical statistics employs methods such as *principal component analysis* (PCA) to describe the whole data set through linear combinations of its variables. These *dimensionality reduction* methods are useful because they permit us to compress a data set to its most important variables, measured e.g. in terms of their variance. We will re-encounter dimensionality reduction methods in Chapter 7, where we analyse several embeddings of high-dimensional data and visualize both their local and global errors.

2.2 Multivariate visualizations

There are a few methods that permit the direct visualization of multivariate data. Among the classical approaches in the visualization community are *parallel coordinate plots* (PCPs), introduced by Inselberg [210], and *scatterplot matrices* (SPLOMs), pioneered by Chambers et al. [86] under the name 'draftman's display'. Figure 2.1 and Figure 2.2 shows examples of these two visualizations when being applied to the 'Iris flower' data set, which we will thoroughly analyse using different clustering algorithms in Chapter 9.

Both approaches work reasonably well for approximately ten variables, but they inevitably have their drawbacks. PCPs, for example, do not have a well-defined axis ordering. Given d dimensions, there are $d!$ different axis arrangements. Ankerst et al. [9] showed that finding the best arrangement—with respect to certain similarity measures, for example—is an NP-complete problem. It is common to employ heuristics, such as the ones introduced by Yang et al. [398], when working with PCPs in practice. Another issue with PCPs is that they work less well for nominal variables or ordinal variables. Likewise, too many observations may result in visual clutter. One strategy against this is to employ filtering [154] or aggregation [156] techniques. Due to their flexibility, PCPs remain an active research topic. Heinrich and Weiskopf [197], for example, extended them to continuous data and showed that these *continuous parallel coordinates* contain less misrepresentations than traditional PCPs.

SPLOMs share some disadvantages with PCPs. Primarily, they scale quadratically with the number of variables—and not all combinations of scatterplots are informative. To discover informative combinations of variables, Friedman and Tukey [171] propose a *projection pursuit* algorithm that looks for 'interesting' projections using certain quality measures. To find these projections more rapidly in scatterplot matrices, we may use the *scagnostics* approach.

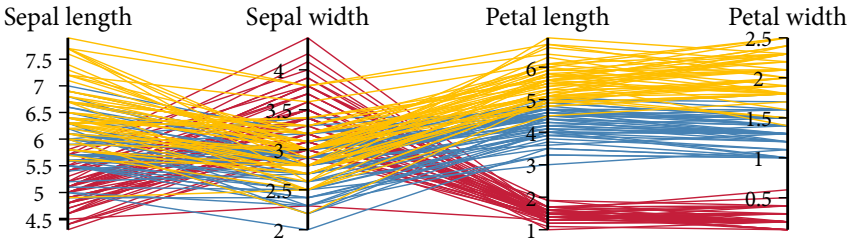


Figure 2.1: An example of a PCP. Every point in the multivariate data set becomes a line in the visualization, while the different axes are parallel to each other. Colours indicate the species of a flower. We can readily see that *I. setosa* flowers ● have smaller petal lengths and petal widths than *I. versicolor* ● or *I. virginica* ●.

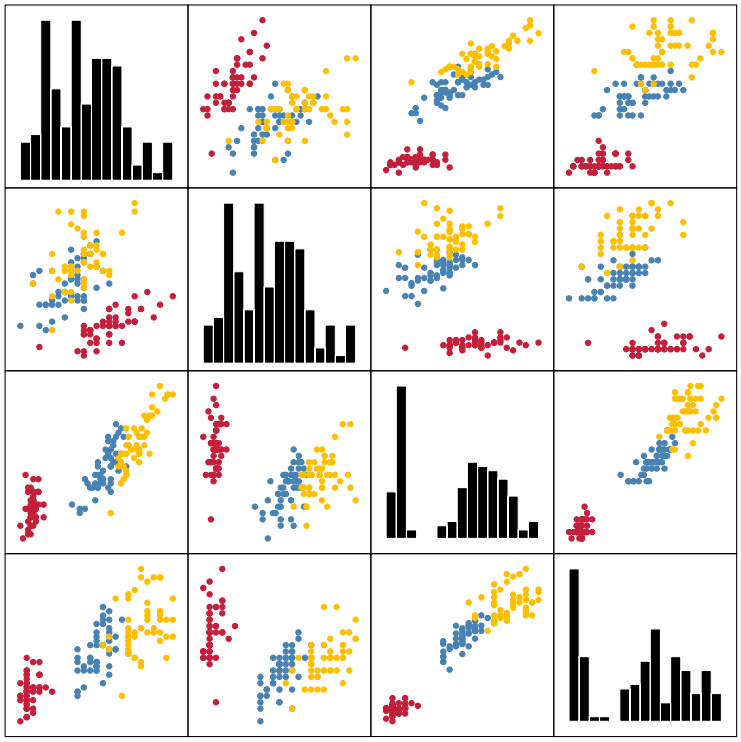


Figure 2.2: An example of a SPLOM. Every non-diagonal element in the matrix is a scatterplot that depicts a certain projection of the data. The diagonal plots often contain additional visualizations such as histograms that display the distribution of the corresponding feature. Points have been coloured according to their species. We can see that there is a pronounced split in some projections between flowers of the species *I. setosa* ● and the two species *I. versicolor* ● and *I. virginica* ●. This visualization also shows that the remaining two species cannot be easily separated by their attribute values. We shall return to this point in Chapter 9, in which we analyse the performance of different clustering algorithms on this data set.

This is a neologism for the *sca*tterplot diag*nostics* by Wilkinson and Wills [392], who define different interestingness measures for rating a given projection. Elmqvist et al. [155] build on this concept by providing interaction concepts for scatterplot matrices that support spatial navigation between different projections. Lehmann et al. [240] approach the scaling problem by pre-processing a SPLOM prior to visualizing it.

Another technique for visualizing certain types of data is the *multivariate heatmap*. It has seen extensive use in the context of bioinformatics, for example. The basic idea of a heat map is to assign colours to the values in different attributes and organize them in a display of columns. Weinstein et al. [390] showed that this approach may be very effective when combined with hierarchical clustering. Even a decade later, it ranks among the most powerful visualization tools for messenger RNA and microRNA expression, protein expression, and gene expression data [389].

Certain types of multivariate data also permit a direct visualization. If all variables of the observations are nominal, one may use *mosaic plots* [190] to visualize their proportions. Likewise, if hierarchical data are to be presented, *tree maps* [336] are a very powerful visualization technique. Originally meant for displaying rectangular regions, they have since been expanded to different shapes [22]. For document collections or texts, the *word cloud* [349, Chapter 3] or ThemeRiver [193] visualizations have proved useful. However, because this thesis deals with generic multivariate data, these visualization techniques are not applicable.

All multivariate visualization approaches are prone to exhibit much clutter if either the number of observations or the number of variables starts to increase. Peng et al. [290] present a generic framework that mitigates this issue.

2.3 Glyph-based visualizations

Multivariate data may also be visualized using special glyphs. One well-known approach involves *star plots* [86], where individual variables are arranged radially and different 'spokes' encode their respective values. This visualization, which is also referred to as *star glyph* or *radar chart*, is simple but powerful because it makes use of the ability of humans to quickly distinguish between different shapes. Figure 2.3 depicts an example of a star plot for the 'Iris flower' data set. However, star plots do not scale well with respect to the number of variables that can be depicted. Likewise, the amount of visual clutter quickly increases and different variable types in the sense of Stevens [350] may not be easily shown.

Star glyphs—or variants of them—are nevertheless often employed to create information-rich displays. The DataMeadow by Elmqvist et al. [157], for example, helps explore even large multivariate data sets. Similarly, the 'stick figure' icons pioneered by Pickett and Grinstein [293] enable the creation of plots that permit the rapid overview of multivariate data as long as

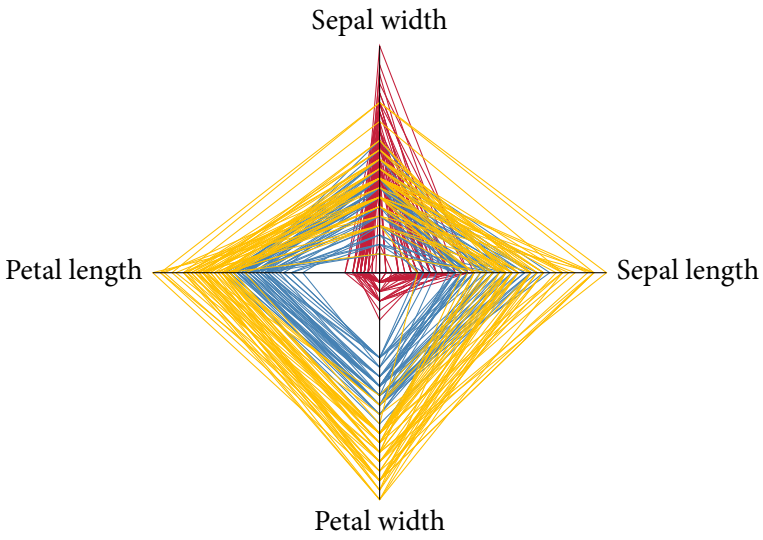


Figure 2.3: An example of a star plot. Here, every high-dimensional data point forms a closed band. The star plot can thus also be seen as a PCP whose first and last axis ave been glued together. Using species-based colours, we again observe that *I. setosa* ● may be easily separated from *I. versicolor* ● and *I. virginica* ●.

the number of variables does not exceed a certain amount. With a growing number of observations, glyph placement strategies become increasingly important and complex to handle [383]. We will use star glyphs in various contexts within this thesis in order to augment existing visualizations.

2.4 Dimensionality reduction methods

A somewhat orthogonal technique to the direct visualization of multivariate data is employed by *dimensionality reduction* algorithms. Briefly put, these algorithms attempt to search for low-dimensional structures in the multivariate data space and visualize them directly. A classical example is given by Tenenbaum et al. [359], who showed that their Isomap algorithm is capable of unrolling a manifold that is intrinsically planar, whereas traditional dimensionality reduction techniques fail to do so. Likewise, *multidimensional scaling* (MDS) is commonly used to obtain low-dimensional embeddings of multivariate data. It is one of the few techniques that is capable of working solely with a matrix of pairwise distances between the data points. Multiple variations of the original algorithm exist; see Borg and Groenen [49] for an excellent overview. The *t-distributed stochastic neighbour embedding* (t-SNE) algorithm [256], by contrast, is one of the few algorithms specifically geared towards yielding high-quality visualizations in low dimensions. Dimensionality reduction methods are not restricted to

generating embeddings that are interpretable by users, though. Often, they are used to 'compress' a set of multivariate observations. Data sets in computational linguistics, for example, typically have several hundred variables, of which only dozens are relevant.

The crux with dimensionality reduction methods is that they always yield *some* answer. Determining the quality or the 'suitability' of a given embedding, however, may become very complex—in fact, it turns out that existing methods have shortcomings, some of which can be fixed by the methods presented in Chapter 7.

2.5 Projection-based visualizations

Some approaches for visualizing multivariate data employ methods from data mining, coupled with projection-based approaches. Liu et al. [252], for example, use subspace clustering [250] to find low-dimensional structures in the data. Each of the detected subspaces is then visualized using a 2D projection. Similarly, Nam and Mueller [281] extract different subspaces using a map metaphor. Interesting subspaces are referred to as 'sights' and the user may plan a 'trip' connecting different sites. Other approaches generate useful projections using statistical methods. The 'Grand Tour' algorithm by Asimov [13], for instance, chooses a sequence of subspaces that is dense in the set of all two-dimensional subspaces. Over the years, these ideas have been refined to include projections that show certain patterns [241], or to aid in outlier detection [24]. For high-dimensional data, even a *random projection* may be shown to yield useful results, both in theory [23] and in practice [47].

The methods in this thesis are complementary to these approaches. In particular in the second part of this thesis, we will use topological methods in order to obtain feature-based descriptions of data. We will then use different projection-based visualizations for augmenting these descriptions.

2.6 Morse theory

When analysing a set of multivariate observations, it is often assumed that they are discrete samples of a continuous manifold of some dimension. This is also known as the 'manifold hypothesis'. The manifold hypothesis is especially prevalent when natural phenomena are being analysed. The reason for choosing manifolds as a model is that they afford a smooth mathematical structure. However, there is no single algorithm for verifying or refuting the manifold hypothesis. If a certain manifold is suspected to be underlying a given data set, an algorithm of Narayanan and Mitter [282] is capable of determining how many samples are required for verifying the hypothesis. In a similar context, Niyogi et al. [285] show how to

'learn' the *homology groups*—a concept which we will discuss at length in Chapter 3—of an unknown manifold with high confidence.

Although manifolds are well-understood in a mathematical sense [236, 238], this understanding does not necessarily result in an improved visualization. Instead of analysing the manifold *directly*, mathematicians aim to describe it by analysing the behaviour of certain functions that are defined *on* it. Somewhat surprisingly, this results in a precise and expressive description of the underlying object. This observation was one of many seminal insights by Marston Morse [273], leading ultimately to the creation of what we today consider to be *Morse theory*. Morse theory, in its modern formulation by Milnor [270], concerns itself with the behaviour of non-degenerate smooth functions on manifolds. It turns out to be possible to relate the critical points—the singularities—of such a function to the homotopy type of the underlying manifold. In order to see how this abstract notion can be used to visualize multivariate data, we first restrict ourselves to analysing changes in the connectivity of their level sets.

Definition 2.1 (Level set). Let $\mathbb{D} \subseteq \mathbb{R}^d$ be a domain and $f\colon \mathbb{D} \to \mathbb{R}$ be a scalar-valued function. Given a value $y \in \mathbb{R}$, a *level set* is the pre-image of f,

$$\mathcal{L}_y(f) := f^{-1}(y) := \{x \in \mathbb{D} \mid f(x) = y\}, \tag{2.2}$$

which is allowed to be empty. If $\mathcal{L}_y(f)$ is not empty, each of its connected components is referred to as a *contour*.

Similarly, we may also define *superlevel sets* and *sublevel sets* of a function. These may be thought of as 'filling the domain with water'.

Definition 2.2 (Sublevel & superlevel set). Let $\mathbb{D} \subseteq \mathbb{R}^d$ be a domain and $f\colon \mathbb{D} \to \mathbb{R}$ be a scalar-valued function. Let $y \in \mathbb{R}$ be a threshold. The *sublevel set* $\mathcal{L}_y^-(f, y)$ and the *superlevel set* $\mathcal{L}_y^+(f, y)$ are the pre-images of f with points whose function value is either below or above the selected threshold:

$$\mathcal{L}_y^-(f, y) := \{x \in \mathbb{D} \mid f(x) \leq y\} \tag{2.3}$$

$$\mathcal{L}_y^+(f, y) := \{x \in \mathbb{D} \mid f(x) \geq y\} \tag{2.4}$$

Both types of sets are again allowed to be empty. Sublevel and superlevel sets are commonly used in scalar field topology and vector field topology [195]. We shall re-encounter these different sets in Chapter 4, for example, when calculating persistent homology.

Figure 2.4 illustrates the concepts of level sets, sublevel sets, and superlevel sets. The level sets of a function on a manifold induce a *quotient topology* [53, pp. 39–44] by identifying two

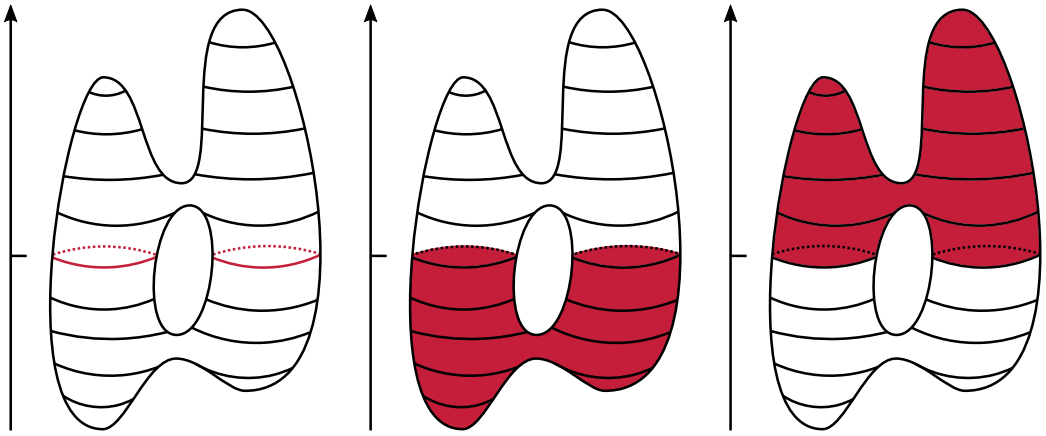


Figure 2.4: An illustration of level sets, sublevel sets, and superlevel sets. In this example, the level set contains two different contours, while the sublevel set and the superlevel set only have a single connected component.

points x and y if and only if they are in the same connected component of some level set of f. If we identify points using this equivalence relation, we obtain the *Reeb graph*. Figure 2.5 depicts a simple Reeb graph whose domain is a 2-manifold. Reeb graphs have been successfully employed for shape modelling [361] and shape analysis, either by visually comparing different graphs [43] or by defining similarity metrics [45] for their semi-automated comparison.

The Reeb graph may be computed very efficiently. Doraiswamy and Natarajan [134] present an algorithm that permits the calculation of the Reeb graph for d-dimensional manifolds with a complexity of $\mathcal{O}\big(n \log n \cdot (\log \log n)^3\big)$, where n is the number of triangles in a triangulation of the manifold. For 2-manifolds, an algorithm of Cole-McLaughlin [108] even manages to obtain a complexity of merely $\mathcal{O}(n \log n)$, making the Reeb graph highly-scalable.

Persistent homology, the main method of this thesis, may be considered a 'superset' of the Reeb graph and related methods. It is not restricted to level set analysis and furthermore, it includes higher-dimensional connectivity information. From the perspective of quantitative data analysis, we shall see that persistent homology permits more mature metrics with known stability properties, whereas efficiently-computable metrics for Reeb graphs are still an open problem [28] and the topic of current research.

If the domain $\mathbb{D}$ of a Reeb graph is simply-connected, i.e. it is path-connected and any path can be contracted to a point, the graph becomes a tree and is referred to as the *contour tree*. Contour trees are often used in the context of scalar field topology in order to support the extraction and simplification of isosurfaces [80]. Using an algorithm by Carr et al. [79], they may be computed in $\mathcal{O}(n \log n + N\alpha(N))$ time, where n is the number of vertices, N is the number of simplices, and $\alpha(\cdot)$ is the extremely slow-growing inverse of the Ackermann

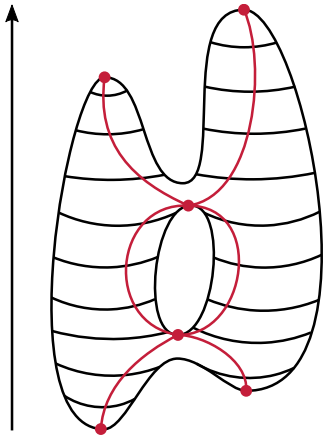


Figure 2.5: An illustration of a Reeb graph. The graph is drawn on top of the manifold whose level sets it describes. Its vertices correspond to the critical points of the height function. Edges signify paths along which the topology of the level set does not change.

function. This is a very efficient construction because N is typically of the order of n, i.e. $N = \theta(n)$. The visualization of contour trees turns out to be challenging. Pascucci et al. [288], for example, use the metaphor of an 'orrery' to produce a 'toporrery' that is drawn radially. Heine et al. [196] developed an algorithm that draws contour trees according to different æsthetic criteria. It is capable of producing readable representations even for larger contour trees with hundreds of branches.

The concepts of the contour tree or the Reeb may be applied to obtain auxiliary representations of scalar functions on multivariate data. A particularly powerful metaphor is the *topological landscape*. Originally pioneered by Weber et al. [388], it has since been expanded to ensembles of scalar functions [191]. The basic idea of these approaches is to generate a landscape in 3D whose contour tree—or Reeb graph, or merge tree, or split tree—coincides with the contour tree of the original data. Since humans are better at spatial reasoning about objects they know, the landscape helps them compare different functions rapidly. Oesterling et al. [286] refined and generalized the original landscape metaphor by exploring density estimates of high-dimensional point clouds. In a follow-up publication, Oesterling et al. [287] refined their approach and concentrated only on *landscape profiles*, i.e. two-dimensional projections of the merge tree structure. This visualization was used to analyse clusters and other substructures in multivariate data.

A more holistic description of multivariate data—still in the context of Morse theory—is obtained by calculating the Morse–Smale complex. The *Morse–Smale complex* [344] is a combinatorial segmentation of the domain into regions of homogeneous gradient flow. This decomposition was represented as a 'spine' in the plane by Correa et al. [116] to obtain simpler descriptions of scalar fields. Gerber et al. [176] use a simplified representation of

the Morse–Smale complex in order to permit the exploration of high-dimensional scalar functions. In lower dimensions, the Morse–Smale complex may be visualized directly. The challenge thus lies more in making the calculations scalable. Due to efficient algorithms [184], approaches based on the Morse–Smale complex are often employed for analysing simulation data, both for large-scale simulations in inertial confinement fusion [55], as well as micro-scale simulations for nanosphere battery materials [185]. Again, there are numerous overlaps between this established usage of Morse theory in visualization and persistent homology. However, persistent homology is applicable even if the domain is not connected. In addition, it permits the description of higher-dimensional features in data, whereas the Morse–Smale complex can only be easily calculated in lower dimensions [144, 146].

In recent publications, efforts have been made to extend concepts from Morse theory to the analysis of multivariate functions. The theoretical basis for this is the *Reeb space* [145], which may be seen as the higher-dimensional equivalent to the Reeb graph. Building on this, Carr and Duke [76, 77] developed the *joint contour net* (JCN), which uses a graph-based approach to show changes in multiple variables. The JCN was already successfully used in analysing complex nuclear scission data sets [137], which have not been amenable to ordinary analysis methods. Current research focuses on generalizing the JCN to multivariate fields with more than two variables [78].

2.7 Feature-based & hybrid methods

Multifield analysis is related to multivariate analysis. In this context, data usually have an underlying domain such as a structured grid. Multiple variables then express themselves as scalar fields, leading to the term *multifield visualization* when multiple scalar fields are to be analysed in the same setting.

Multifield analysis permits numerous different viewpoints. An approach by Sauber et al. [323], for instance, considers correlations between individual scalar fields to be interesting features. These are summarized in the *multifield graph*. Jänicke et al. [214] calculate an Euclidean minimum spanning tree (EMST) on multivariate data and use graph layout techniques to obtain a 2D visualization. Users may then explore relationships in the data by interacting with this *attribute cloud*. Hüttenberger et al. [207] proposed a new framework based on multivariate optimization strategies to describe features in multiple scalar functions. This yields a new set of features that may be used to understand and compare the behaviour of different vortex criteria, for example.

Additionally, many methods use ideas from persistent homology and Morse theory, without being firmly footed in either one of these fields. The Mapper algorithm [342], for example, combines hierarchical clustering with basic data structures from computational topology

to form a low-dimensional simplicial complex. This complex describes connectivity relations between different data points. It is commonly visualized using graph drawing methods. MAPPER is very effective in describing complex network data such as folding pathways [400], phylogenetic trees [87], and 'omics' data [283]. However, the numerous parameters and quantization parameters of the MAPPER algorithm make assessing its stability complicated. Formal theorems are hence an ongoing area of research [81].

3 Algebraic topology

Most of the methods that are used and developed in this thesis are based on algebraic topology to some extent. This chapter introduces and motivates the required concepts. We will briefly expand on *simplicial homology*, a method for calculating certain invariants of topological spaces. These ideas lay the foundation for *persistent homology*, a powerful data analysis tool, which we will describe at length in Chapter 4.

✦

Algebraic topology is one of the more recent branches of mathematics. It is concerned with the identification, description, and discrimination of high-dimensional objects. In contrast to the more geometrically-inclined field of differential topology, methods from algebraic topology rely solely on the connectivity of a space. Geometry is left out for reasons we will address later on. While the connection is not immediately obvious, the problems of algebraic topologists and visualization researchers share some commonalities. Given high-dimensional data, we attempt to visualize it in some way, such that it is possible to see intrinsic structures. Our visualization needs to be *efficient* in the sense that it should be obtainable by reasonable computational means, but also *discriminative* in the sense that it should permit us to distinguish some forms of data from other forms of data. Similarly, algebraic topologists want to understand and distinguish high-dimensional objects. Since human intuition fails beyond even three or four dimensions, topologists approach the problem by identifying intrinsic properties of high-dimensional spaces that do not change under some transformations. These properties are referred to as *invariants*. Invariants are similar to our visualizations. To be useful, they need to be efficiently computable and sufficiently discriminative at the same time.

Topology reduces the complex task of classifying and discerning high-dimensional objects by disregarding the effects of certain transformations. A reasonable set of transformations that may be ignored is the set of affine transformations, such as movements and rotations. In algebraic topology, we ignore an even larger class of operations, namely all operations that are described by homeomorphisms.

Definition 3.1 (Homeomorphism). A function $f\colon \mathbb{X} \to \mathbb{Y}$ between two topological spaces $\mathbb{X}$ and $\mathbb{Y}$ is a *homeomorphism* if f is bijective, continuous, and its inverse $f^{-1}\colon \mathbb{Y} \to \mathbb{X}$ is continuous as well.

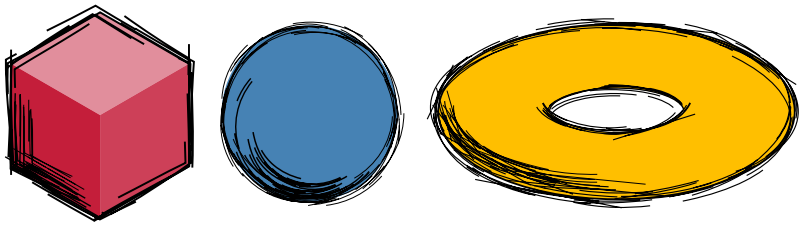


Figure 3.1: Examples of homeomorphic and non-homeomorphic objects. The cube (left) can be deformed into a sphere (middle). Both objects are non-homeomorphic to the torus (right), though, because it is impossible to obtain the hole of the torus without cutting the sphere.

In other words, a homeomorphism is any transformation that involves stretching, translating, and bending—but neither cutting nor gluing. It is reasonable to demand that characteristic properties of a space must not change under this type of transformation. Figure 3.1 shows several examples of homeomorphic and non-homeomorphic objects. The classification up to homeomorphism might seem weak with too much leeway to be highly-discriminative. For the properties that topologists aim to study, this classification is sufficient, though. Later on, when we introduce persistent homology, we will also see that while this method is based on concepts of algebraic topology, it has a much higher discriminative power. The discrete and multi-scale nature of real-world data makes it possible for persistent homology to detect more variation in data sets.

3.1 Topological spaces & their invariants

Topologists are interested in studying *topological spaces*. In this thesis, we think of a topological space as a set of points, usually sampled from some $\mathbb{R}^n$, with a neighbourhood relation. The neighbourhood relation permits us to query any point about its connectivity. Figure 3.2 illustrates this concept. This definition is purposefully vague and excludes many other objects in order to permit a better exposition of the relevant theoretical concepts. A topological space does not presuppose the existence of a metric, such as the Euclidean distance. It is thus fundamentally different from the metric spaces we are used to. Later on, we will extend the definition of a topological space when we have some means of ascertaining the dissimilarity of points in our data.

A very simple topological invariant, for example, is given by the dimension of the space. To distinguish $\mathbb{R}^n$ from $\mathbb{R}^m$, we only need to look at whether $n \neq m$. The dimensionality of the input space is of course not a great invariant and will be insufficient in most cases—especially for real-world data where the 'correct' dimension is often unknown. *Simplicial homology*, which we shall focus on in this chapter, is a more balanced invariant. The motivation for simplicial homology is to use the number of 'holes' in a topological space as an invariant.

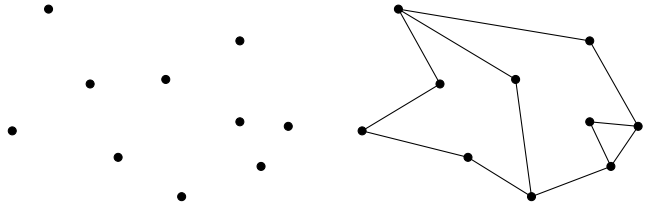


Figure 3.2: The notion of a topological space is less intuitive than that of a metric space. To the left, we have a set of points from $\mathbb{R}^2$. The right part shows one particular topological space of the points. Edges indicate neighbours but those neighbours do not necessarily correspond to our sense of proximity.

Returning to Figure 3.1, two of the depicted objects—the sphere and the cube—do not have a hole. The torus, by contrast, has at least one hole. Since homeomorphisms do not permit tearing, we will never be able to transform the cube or the sphere into a torus. Thus, they are fundamentally different. The idea of using connectivity information to classify spaces has already found an application in the *Betti numbers* [192, p. 130] of a space—we shall return to this concept shortly. Simplicial homology is effectively computable, but requires a somewhat complex algebraic set-up, which the subsequent sections are going to cover.

3.2 Simplicial homology

To identify holes in a space, we first need a notion of its connectivity. For this, we require the topological space being described by a *simplicial complex*, one of the basic building blocks in algebraic topology. A simplicial complex is a data structure that explains how to 'glue together' a topological space from points, edges, triangles, tetrahedra, and their higher-dimensional generalizations. Simplicial complexes thus serve as 'blueprints' of a space. The points, edges, and other entities that make up a simplicial complex are known as *simplices*. There are other complexes, such as *CW complexes* [192, p. 5 ff.], but we prefer a description in terms of simplicial complexes because they are significantly easier to handle.

In the following, we shall define simplicial complexes and simplices in a combinatorial way because we want to perform calculations on a computer with them later on. However, it is also possible to define everything in a geometrical manner—see Bredon [53, pp. 245–250], for example. We start by describing simplices and simplicial complexes.

Definition 3.2 (Abstract simplex). Given a family of sets, any subset of cardinality $k+1$ is called a *k-simplex*. In a graph-theoretic context, we may think of 0-simplices as vertices, 1-simplices as edges, and 2-simplices as triangular faces. This intuition generalizes to higher dimensions as well.

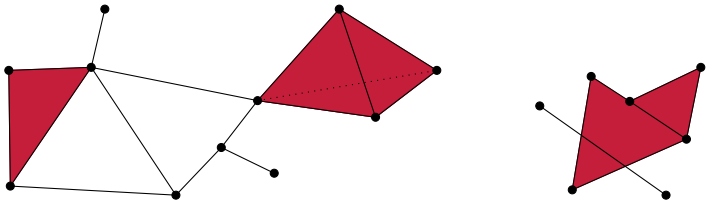


Figure 3.3: An abstract simplicial complex (left) and a set of simplices (right) that is not an abstract simplicial complex because the two triangles do not intersect on a common face. By requiring all 'faces' of a simplex to be part of the simplicial complex as well, algorithms can easily traverse the boundaries and need not keep track of any missing parts.

Definition 3.3 (Abstract simplicial complex). A family of sets K with a collection of subsets L is called an *abstract simplicial complex* if:

1. $\{v\} \in \mathrm{L}$ for all $v \in \mathrm{K}$. The sets of the form $\{v\}$ are the *vertices* of the simplicial complex. We shall also denote them as $\operatorname{vert} \mathrm{K}$.

2. If $\sigma \in \mathrm{L}$ and $\tau \subseteq \sigma$, then $\tau \in \mathrm{L}$. We refer to τ as a *face* of σ and to σ as a *coface* of τ.

Intuitively, the first constraint is required to ensure that the simplicial complex contains all 0-simplices, which we refer to as vertices because they only consist of a single point (or rather its index). The second constraint enforces that the subsets, i.e. the simplices, only intersect along shared boundaries. Figure 3.3 shows a geometrical representation of both properties.

There are certain subsets—or rather *subcomplexes*—in a simplicial complex that will be relevant in subsequent chapters. The most important one is the idea of a k-skeleton of a simplicial complex.

Definition 3.4 (k-skeleton of a simplicial complex). Given a simplicial complex K, the set of all simplices with dimension $\leq k$ forms a valid simplicial subcomplex. This subcomplex is called the *k-skeleton* of K. The 1-skeleton of a simplicial complex, for example, only contains 0-simplices and 1-simplices. Hence, it is a graph.

Even though we restrict ourselves to topological spaces that are representable as a simplicial complex, we still have a broad range of spaces to choose from. This is due to a deep theorem in algebraic topology, which we cite for completeness, without defining all terms.

Theorem 3.5 (Simplicial approximation theorem). Every smooth manifold admits one essentially unique compatible piecewise linear structure that is equivalent to a combinatorial triangulation, i.e. a simplicial complex.

Proof. See the seminal publications of Cairns [65] or Whitehead [391] for more details. ∎

Since we commonly assume that real-world data is a discrete approximation of essentially continuous phenomena, smooth manifolds are precisely the objects we are interested in. From the perspective of computer science, the simplicial complex representation is also very appealing for at least two reasons. First, there are numerous smart data structures for storing a simplicial complex efficiently [15, 123, 124]. Second, the calculation of simplicial homology for simplicial complexes boils down to matrix calculations.



We now have a concept for describing the connectivity of a topological space. However, we also need a better notion of boundaries and holes. A k-dimensional hole in the sense of algebraic topology is a part of a topological space at which a $(k+1)$-dimensional ball could be attached (subject to a homeomorphism). The ball then 'closes up' the hole. This description corresponds to our intuition: Connected components are 0-dimensional holes, because they can be closed up by inserting a 1-ball (an edge). Tunnels are 1-dimensional holes, because we can fix them by inserting a 2-ball (a disk). Likewise, voids are 2-dimensional holes, because they can be fixed using 3-ball. Unfortunately, this is where our intuition stops—algebraically, higher-dimensional cavities can be fixed in the same way, though.

In order to find these holes, we need to be able to calculate boundaries in a simplicial complex. We may then look for those parts of a simplicial complex that do not have a boundary. This idea requires the definition of a *boundary operator* on a simplicial complex, as well as a formal structure—the *chain groups*—that describe the boundary relations between simplices of different dimensions. We first formalize the algebraic structure of simplices within a simplicial complex.

Definition 3.6 (Chain group of a simplicial complex). Given a simplicial complex K, the p^{th} chain group C_p of K is defined by all *formal linear combinations*, i.e. formal sums, of p-simplices in the complex. All elements of C_p are thus of the form $\sum_j \sigma_j$ for $\sigma_j \in \mathrm{K}$. The coefficients in these linear combinations are only allowed to be 0 or 1, meaning that a simplex is either part of a linear combination or not.

The chain group permits us to define a group structure over simplicial complexes. In particular, it will permit us to obtain a well-defined notion of concepts such as *boundaries* and *cycles*. Restricting coefficients to 0 or 1 amounts to working over the cyclic group of two elements, $\mathbb{Z}/2\mathbb{Z}$ or $\mathbb{Z}_2$. While simplicial homology (and likewise, persistent homology) works equally for other coefficient sets [192, pp. 153–155], the calculations become more involved. Zomorodian [406, pp. 56–57] remarks that for most real-world data sets, $\mathbb{Z}_2$ is the perfect choice. However, the methods and visualizations presented in this thesis do not depend on the choice of coefficients and can be applied in more general situations.

Definition 3.7 (Simplicial chain). We call the elements of the p^{th} chain group C_p *simplicial chains*. Each simplicial chain is thus a linear combination of p-simplices. Since C_p is a group, we may calculate the addition of two different simplicial chains. Given two simplicial chains a and b, their addition over $\mathbb{Z}_2$ coefficients is equivalent to calculating their *symmetrical difference*:

$$a + b = (a \cup b) \setminus (a \cap b) \tag{3.1}$$

In other words, $a + b$ contains all those simplices that only occur in either a or b. The choice of $\mathbb{Z}_2$ coefficients is very advantageous here, because the symmetrical difference can be implemented efficiently.

Having defined a group operation, given by the symmetrical difference between two simplicial chains, the chain group now has the mathematical structure of a group. Prior to deriving more properties about chain groups, we formalize their properties.

Lemma 3.8. The chain group is an abelian group with the set operation given by the symmetrical difference between simplicial chains.

Proof. The chain group is closed under the group operation. The 'addition' of two simplicial chains will thus always result in another valid element of the chain group. Furthermore, the symmetrical difference is associative and commutative by inspection—the order in which we calculate the symmetrical difference is irrelevant. Likewise, the group operation has an identity element—the empty simplicial chain—whose addition to any simplicial chain does not change it. Since we are calculating in $\mathbb{Z}_2$ coefficients, each simplicial chain is its own inverse. The chain group thus satisfies all axioms required for an abelian group. ∎

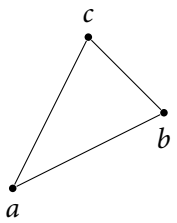
We are now able to define the boundary of a simplex as a linear function from one chain group to another.

Definition 3.9 (Boundary homomorphism). Given a simplicial complex K, the p^{th} boundary homomorphism is the map that assigns each simplex $\sigma = \{v_0, \ldots, v_p\} \in \mathrm{K}$ to its boundary, which is a simplicial chain:

$$\partial_p \sigma \coloneqq \sum_i \{v_0, v_1, \ldots, \hat{v}_i, \ldots, v_k\} \tag{3.2}$$

In the equation above, $\hat{v}_i$ indicates that the set does *not* contain the i^{th} vertex. The function $\partial_p \colon C_p \to C_{p-1}$ is a homomorphism between the chain groups.

We have claimed that the function defined above is a homomorphism between groups. To verify this, it suffices to show that $\partial_p(\sigma + \tau) = \partial_p\, \sigma + \partial_p\, \tau$ for simplices σ and τ. In other words, we need to show that ∂_p is *compatible* with the group operation we defined above.



The boundary of the triangle is non-zero. We have $\partial_2\{a,b,c\} = \{b,c\} + \{a,c\} + \{a,b\}$. The set of edges, on the other hand, does not have a boundary, i.e. $\partial_1(\{b,c\} + \{a,c\} + \{a,b\}) = \{c\} + \{b\} + \{c\} + \{a\} + \{b\} + \{a\} = 0$, because the simplices cancel each other out.

Figure 3.4: Calculating the boundaries of a 2-simplex (a triangle) and its 1-simplices (edges). The boundary of a boundary is always zero; this property is summarized as the fundamental lemma of simplicial homology.

Lemma 3.10. We have $\partial_p(\sigma + \tau) = \partial_p\,\sigma + \partial_p\,\tau$ for two arbitrary simplices σ and τ. In other words, ∂_p is a homomorphism between groups.

Proof. If σ and τ have no faces in common, it does not matter in which order the calculations are being performed. In case σ and τ share some faces, all shared faces will disappear when calculating $\sigma + \tau$. When calculating the boundary of each common face, on the other hand, common faces will result in duplicate boundaries. Hence, their addition again cancels out. ■

Geometric view While seemingly abstract, this definition of the boundary operator captures our notion of a boundary correctly. Figure 3.4 illustrates the boundary calculation for a triangle and its edges. The figure seems to imply that boundaries do not themselves have a boundary—which matches our intuition. This property of the boundary homomorphism is fundamental for simplicial homology. The cycle and boundary groups are connected by the following fundamental lemma.

Lemma 3.11 (Fundamental lemma of simplicial homology). For all p, we have $\partial_{p-1} \circ \partial_p = 0$, i.e. the function that assigns every simplicial chain to zero.

Proof. We have:

$$\partial_{p-1}\partial_p\sigma = \partial_{p-1}\sum_i \{v_0,\dots,\hat{v}_i,\dots,v_p\} \tag{3.3}$$

$$= \sum_{i<j}\{v_0,\dots,\hat{v}_i,\dots,\hat{v}_j,\dots,v_p\} + \sum_{j>i}\{v_0,\dots,\hat{v}_j,\dots,\hat{v}_i,\dots,v_p\} \tag{3.4}$$

Since the two partial sums in the preceding equation contain the same terms, they cancel each other out if we calculate with $\mathbb{Z}_2$ coefficients. ■

The fundamental lemma permits us to define a chain complex, in which we can relate all the simplices of a simplicial complex. Later on, we will use the chain complex to define algebraic operations.

Definition 3.12 (Chain complex of a simplicial complex). The chain complex of an n-dimensional simplicial complex K is the sequence of chain groups, connected with the corresponding boundary homomorphisms:

$$0 \xrightarrow{\partial_{n+1}} C_n \xrightarrow{\partial_n} C_{n-1} \xrightarrow{\partial_{n-1}} \dots \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0 \tag{3.5}$$

Since every object within the chain complex is an abelian group according to Lemma 3.8, the notion of subgroups in each C_p is well-defined.

Returning back to Figure 3.4, we need to examine our notion of a 'hole'. A useful definition is to declare a simplicial chain to be a hole if it does not have a boundary. Intuitively, we would say that the simplicial complex in the figure has a hole if it does not contain the triangle $\{a, b, c\}$. In that case, we would consider the simplicial chain created by the boundary of the triangle to describe a hole. The existence of a hole hence depends on both the simplices in dimension p as well as the simplices in dimension $p+1$. To formalize this, we shall take a look at two important subgroups of C_p, namely the *cycle group* Z_p (which takes its name from the German word 'Zykel'), and the *boundary group* B_p.

Definition 3.13 (Cycle and boundary groups). Given a chain group C_p, the p^{th} cycle group is defined by

$$Z_p := \ker \partial_p, \tag{3.6}$$

meaning that Z_p contains all those p-simplices (and simplicial chains) that do not have a boundary. Thus, an element of Z_p is also called a p-cycle. In the same vein, the p^{th} boundary group is defined by

$$B_p = \operatorname{im} \partial_{p+1}, \tag{3.7}$$

meaning that it contains all the boundaries of $(p+1)$-dimensional simplices. Note the shift in dimensions here—it is required because a higher-dimensional simplex needs to exist in order for the chain to *be* a boundary! Simplicial chains in B_p are thus also called *bounding cycles*.

Subgroup relation As a consequence of the fundamental lemma, any p-boundary is also a p-cycle. The p^{th} boundary group B_p is hence a subgroup of the p^{th} cycle group Z_p, i.e. $B_p \subseteq Z_p$. Why do we require the distinction between cycles and boundaries? As we have seen in Figure 3.4, we must not consider a simplicial chain to be a hole when it constitutes the boundary of a higher-dimensional simplex. Hence, to separate these 'fake' holes from 'real' holes, we need to remove the higher-dimensional boundaries. This amounts to calculating a *quotient group* [11, pp. 66–70] of C_p, and leads us to the definition of a homology group.

Definition 3.14 (Homology group). Given a chain group C_p and its subgroups Z_p and B_p, the p^{th} homology group is defined as

$$H_p := Z_p / B_p = \ker \partial_p / \operatorname{im} \partial_{p+1}, \tag{3.8}$$

where the /-operator refers to the quotient group. The resulting group, H_p, consists of *equivalence classes* of cycles. This means that elements in H_p are only defined up to a boundary. In other words, if we have two cycles $a, b \in Z_p$ and $a = b + c$, for some $c \in B_p$, then we will not be able to distinguish a and b in H_p any more. Again, this corresponds to our intuition: If two cycles only differ by elements from the boundary group, and we remove said boundaries using the quotient operation, the elements will effectively be the same.

Since elements in the p^{th} homology group H_p are only defined 'up to a boundary', we shall also refer to them as *homology classes*. This accounts for the fact that elements in H_p are equivalence classes.

Algebraic structure of homology groups The homology group H_p has a rich algebraic structure. Often, we are not interested in this structure but rather desire a short summary of the group. Such a short summary is given by the *Betti numbers*.

Definition 3.15 (Betti numbers). The p^{th} *Betti number* β_p is the algebraic rank of the p^{th} homology group, i.e. $\beta_p := \operatorname{rank} H_p$.

The Betti numbers are commonly used to distinguish different topological spaces from each other. This even works regardless of their dimensionality. For example, referring back to Figure 3.1 on p. 26, a 2-sphere has Betti numbers $\beta_0 = 1$ (a single connected component), $\beta_1 = 0$ (no tunnels), and $\beta_2 = 1$ (a single void that is enclosed by the sphere), while all other Betti numbers are zero. A torus, on the other hand, has Betti numbers $\beta_0 = 1$, $\beta_1 = 2$ (because there are two non-homologous loops on the surface of the torus), and $\beta_2 = 1$ (because it encloses a void just like the sphere). We thus know a sphere to be different from a torus just by looking at their 1-dimensional Betti numbers. Table 3.1 shows the Betti numbers of several simple topological spaces.

This signature property of Betti numbers is well-defined and even holds for the complete homology groups of a topological space. To show that homology groups are a useful topological invariant, we briefly consider what happens when two simplicial complexes X and Y are homeomorphic. We have the following theorem, whose proof is non-trivial [192, pp. 103–133].

Theorem 3.16 (Invariance of simplicial homology). If X and Y are homeomorphic simplicial complexes, their homology groups are isomorphic, i.e. we have $H_p(X) \simeq H_p(Y)$ for all dimensions p.

Space	β_0	β_1	β_2	β_3
Point	1	0	0	0
Circle	1	1	0	0
2-sphere	1	0	1	0
3-sphere	1	0	0	1
Klein bottle	1	2	0	0
Torus	1	2	1	0

Table 3.1: Betti numbers for several common topological objects, calculated with coefficients in $\mathbb{Z}_2$. We can see that every k-sphere satisfies $\beta_j = 1$ only for $j = 0$ and $j = k$.

Limits of homology groups Unfortunately, the converse of the previous theorem is not true. Thus, just from the fact that X and Y have the same homology groups, we cannot conclude that X and Y are homeomorphic. A classical example of this is the class of *homology spheres*. These are spaces that have the same homology groups as spheres, but they are non-homeomorphic to spheres. The *Poincaré homology sphere* [377], for example, is a manifold that is known not to be homeomorphic to the Euclidean 3-sphere.

3.3 Relative simplicial homology

Prior to discussing algorithms for calculating simplicial homology, we briefly discuss the more advanced topic of *relative simplicial homology*. We require this Definition in Chapter 6, where we use it to reduce the geometrical size of simplicial chains.

Relative simplicial homology was introduced after observing that by ignoring certain parts of a topological space, simplicial homology may sometimes become a more powerful description and more easy to calculate. Formally, we may ignore a part of space by calculating quotient groups. In the following, we assume that we are given a simplicial complex K and a subcomplex $\mathrm{L} \subseteq \mathrm{K}$. Instead of the usual description of relative simplicial homology, as given by Hatcher [192, pp. 115–119], for example, we rather describe everything in terms of simplicial complexes. Using the two simplicial complexes, we may now describe relative versions of the algebraic objects we already encountered.

Definition 3.17 (Relative chain group). Let $C_p(\mathrm{K})$ be the chain group of the simplicial complex and $C_p(\mathrm{L})$ be the chain group of the subcomplex. Since $C_p(\mathrm{L}) \subseteq C_p(\mathrm{K})$, taking the quotient group is well defined. We write

$$C_p(\mathrm{K}, \mathrm{L}) := C_p(\mathrm{K}) / C_p(\mathrm{L}) \tag{3.9}$$

for the chain group in which all chains from L are trivial. The relative chain groups are connected just like the ordinary chain groups because the boundary operator respects the subgroup relation.

The fundamental lemma of simplicial homology still applies *after* taking the quotient because it holds *before* taking the quotient, as well. We can thus define relative simplicial homology groups in a similar manner to Definition 3.14. In particular, the relative cycle group and the relative boundary group are defined just like their ordinary counterparts.

Definition 3.18 (Relative simplicial homology group). Given the relative cycle group $Z_p(\mathrm{K}, \mathrm{L})$ and the relative boundary group $B_p(\mathrm{K}, \mathrm{L})$, the relative simplicial homology group is defined as

$$H_p(\mathrm{K}, \mathrm{L}) := Z_p(\mathrm{K}, \mathrm{L}) / B_p(\mathrm{K}, \mathrm{L}), \tag{3.10}$$

where we define the quotient using the induced boundary homomorphism. We deliberately refrain from defining all the relative algebraic constructs in order to avoid confusion; they follow rather naturally from the definitions.

We briefly think about the elements encountered in a relative simplicial homology group. Any element in $H_p(\mathrm{K}, \mathrm{L})$ is represented by a *relative cycle*, i.e. a p-dimensional simplicial chain $a \in C_p(\mathrm{K})$ such that $\partial_p\, a \in C_{p-1}(\mathrm{L})$. Similarly, a relative cycle a is trivial in $H_p(\mathrm{K}, \mathrm{L})$ if and only if it is a *relative boundary*, i.e. it can be written as $a = \partial_p\, b + c$ for simplicial chains $b \in C_{p+1}(\mathrm{K})$ and $c \in C_p(\mathrm{L})$. Written somewhat tongue-in-cheek, we follow the example of Hatcher [192, p. 115] and consider $H_p(\mathrm{K}, \mathrm{L})$ to be akin to 'simplicial homology of K modulo L'.

3.4 Calculating simplicial homology

The simplicial homology groups of a topological space can be calculated in a straightforward manner, provided that the space is described by a simplicial complex. We shall see that the calculation of the homology groups essentially boils down to reducing certain matrices that are induced by the boundary homomorphisms. Munkres [277, pp. 53–61] gives a detailed account of the standard reduction algorithm. The basic idea is to reduce all boundary matrices to their *Smith normal form* (SNF), from which suitable bases for the chain complexes can be read off. Before showing the connection to the homology group, we first define the SNF.

Definition 3.19 (Smith normal form). Let M be an $n \times m$ matrix with at least one non-zero entry over some field $\mathbb{F}$. There are invertible matrices S and T such that the matrix product SMT has the form

$$SMT = \begin{pmatrix} b_0 & 0 & 0 & & \cdots & & 0 \\ 0 & b_1 & 0 & & \cdots & & 0 \\ 0 & 0 & \ddots & & & & 0 \\ \vdots & & & b_k & & & \vdots \\ & & & & 0 & & \\ & & & & & \ddots & \\ 0 & & & \cdots & & & 0 \end{pmatrix}, \tag{3.11}$$

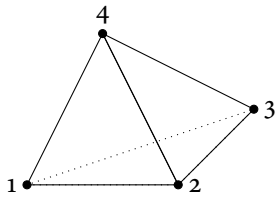
where all the entries b_i satisfy $b_i \geq 1$ and divide each other, i.e. $b_i \mid b_{i+1}$. All b_i are unique up to multiplication by a unit.

Note that this definition is more general and works over any principal ideal domain as well. The SNF may be computed in a similar manner to the standard *Gaussian elimination* [11, pp. 9–19] technique. The major difference is that all entries remain integers while computing the SNF, which is not guaranteed for the Gaussian elimination algorithm. Subsequently, the operation of obtaining the SNF of a matrix is denoted by the $\simeq$ symbol, i.e. we write $M \simeq SMT$.

How may the SNF help us calculate the p^{th} homology group H_p of a given simplicial complex K? We first recall that by Lemma 3.10, ∂_p and ∂_{p+1} are homomorphisms. As such, we may represent them by matrices. Given a lexicographical ordering of simplices in dimension p and $p+1$, we associate ∂_p with a matrix M_p that has as many columns as there are p-simplices in K, and as many rows as there are $(p+1)$-simplices in K. For each column, we write a 1 in all those rows that correspond to simplices that occur in the boundary of the given simplex. For ∂_{p+1}, we apply the same procedure. From the SNF of both homomorphisms, we may read off a complete description of the p^{th} homology group H_p.

Theorem 3.20. The rank of the p^{th} cycle group Z_p is the number of zero columns in M_p. The rank of the p^{th} boundary group B_p is the number of non-zero rows in M_{p+1}.

Proof. The proof is part of a longer argument involving a reduction algorithm for matrices. See Munkres [277, pp. 58–59] for more details. ∎



0-skeleton	$\{1\}, \{2\}, \{3\}, \{4\}$
1-skeleton	$\{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\}$
2-skeleton	$\{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,4\}$

Figure 3.5: A simplicial complex. The complex does not contain the tetrahedron consisting of its four vertices—rather, only the four individual triangles of the faces of the tetrahedron are present. The simplicial complex is thus the smallest triangulation of the 2-sphere and we would expect its homology groups to coincide with those of the sphere.

Since the ranks of abelian groups decomposes similarly to the *rank–nullity theorem* [11, pp. 110–111], we may use the previous theorem to calculate the p^{th} Betti number β_p as

$$\beta_p = \operatorname{rank} Z_p - \operatorname{rank} B_p \tag{3.12}$$

by counting the appropriate rows and columns of the reduced matrices. While this algorithm is relatively simple, it is not the most practical way of calculating simplicial homology. Its base complexity is $\mathcal{O}\left(n^3\right)$ for calculating even a *single* homology group! More efficient schemes are still an area of ongoing research [120, 138]. The ideas of this technique will be useful, though, when calculating *persistent homology*. We shall see that for this purpose, a single reduction of a larger matrix will be sufficient. The following theorem, whose proof is given by Munkres [277, p. 60–61], summarizes the observations in this chapter.

Theorem 3.21. The homology groups of a finite simplicial complex K are *effectively computable*, for example by reducing the boundary matrix of K to its SNF.

An example

In the following, we apply the concepts presented in this chapter and calculate all simplicial homology groups of a given simplicial complex. Suppose that we want to calculate simplicial homology of the simplicial complex K depicted in Figure 3.5. We first calculate the boundary matrix M_0. This is easy because, by definition, all 0-simplices are mapped to zero. The matrix thus reads

$$M_0 = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}, \tag{3.13}$$

meaning that all vertices are sent to zero. The SNF of this matrix is simply the matrix itself.

Applying Theorem 3.20, we thus know that rank $Z_0 = 4$. The boundary matrix M_1 of the 1-dimensional simplices reads

$$M_1 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \simeq \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{3.14}$$

so rank $B_0 = 3$. Knowing these ranks permits us to calculate

$$\beta_0 = \operatorname{rank} Z_0 - \operatorname{rank} B_0 = 4 - 3 = 1, \tag{3.15}$$

which is not surprising because K only has a single connected component. Likewise, from the previous matrix we know that rank $Z_1 = 3$ because there are three zero columns. To calculate β_1, we need to calculate and reduce M_2. We have

$$M_2 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \simeq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.16}$$

by the SNF. Hence, we have rank $B_1 = 3$ because there are three non-zero rows in the SNF. Thus, $\beta_1 = \operatorname{rank} Z_1 - \operatorname{rank} B_1 = 3 - 3 = 0$. We now have sufficient information to calculate β_2. Because the simplicial complex does not contain any higher-dimensional simplices, there are no higher-dimensional boundaries and we have rank $B_2 = 0$. It thus suffices to count the number of zero columns in the reduced matrix to determine that rank $Z_2 = 1$, and therefore $\beta_2 = 1$. In summary, the Betti numbers coincide with that of the 2-sphere and we can see that our simplicial complex is a valid triangulation of this object.

3.5 Discussion

This chapter presented the most important aspects of algebraic topology. We first discussed the concept of invariants for distinguishing between different topological spaces, showing parallels between the goals of algebraic topology and visualization. Following this, we started to focus on *simplicial homology*, an invariant that is computable due to its combinatorial nature. Simplicial homology requires numerous mathematical concepts, which we briefly

discussed and explained. The most important one of these concepts is the notion of a *simplicial complex*, a data structure that makes the algebraic definition of simplicial homology amenable to combinatorial calculations. In the next chapter, we will discuss how to obtain simplicial complexes that capture the connectivity of generic multivariate data sets. Moreover, we will see how to extend simplicial homology to a multi-scale structural descriptor of such data.

4 Persistent homology

In the previous chapter, we have encountered *simplicial complexes*, one of the building blocks for calculating invariants in algebraic topology. Real-world data is not commonly given in such a form. Domain experts often work with high-dimensional point clouds, i.e. a set of vectors from some $\mathbb{R}^n$, where each dimension represents a single attribute [251]. We may assume the *manifold hypothesis* to be true for these data sets, in particular when they describe a continuous phenomenon. As we learned earlier, this does not necessarily give us a usable model of our data. The basic premise of *topological data analysis* (TDA) is the attempt to work with data for which no known manifold model exists. TDA describes the structure of an underlying manifold, provided the manifold hypothesis for the data is true. While the field of TDA has been slowly emerging over the last few years, several techniques for data analysis have been developed. All of them are based on concepts in algebraic and differential topology. An article by Ghrist [178] gives a good overview of different problems and methods.



This chapter describes *persistent homology*, a 'real-world approximation' of simplicial homology. Of all the techniques for TDA, it is the most developed one [142, 149]. Persistent homology offers a rather unorthodox approach towards describing data. It focuses on describing data in terms of what is missing, i.e. in terms of tunnels, holes, voids, and higher-dimensional cavities. While this chapter focuses on methods geared towards obtaining simplicial complexes from high-dimensional point cloud data, neither persistent homology nor the methods in this thesis are restricted to point clouds. In fact, since topological methods only focus on the connectivity of a space, the methods described here are applicable in more general contexts, for example to analyse network data [338] or functional brain connectivity [235].

In the following sections, we will first define different ways of obtaining simplicial complexes from multivariate data. After discussing their approximation properties, we will focus on a particular simplicial complex, the Vietoris–Rips complex. This complex is well-suited to describe the multivariate data. Next, we will discuss the concept of *persistent homology*, which permits us to calculate topological properties of multivariate data. We will first present a novel and simplified algorithm for calculating 0-dimensional persistent homology for functions and arbitrary graphs. This is followed by a discussion of the need for multi-scale descriptions

of topological features in high-dimensional data. Finally, we will describe two different algorithms for calculating persistent homology in arbitrary dimensions. These algorithms are based on the SNF reduction we encountered in Chapter 3. If not mentioned otherwise, all proofs are due to the author.

The remainder of this chapter then deals with introducing concepts related to persistent homology. To this end, we will discuss two standard visualization methods—*persistence diagrams* and *persistence barcodes*—and their properties. Moreover, we will define two different distance measures between persistence diagrams, the *bottleneck distance* and the *Wasserstein distance*. We will describe numerous notions of stability of these distance measures and encounter theorems that state the conditions under which they remain robust with respect to noise. The chapter ends with a comparison of topological and geometrical distances, during which we briefly outline several advantages and beneficial properties. This discussion is an extended version of a brief argument in an earlier publication of the author [311].

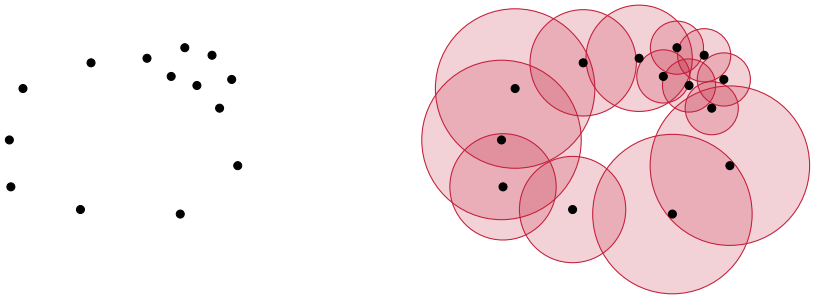
4.1 Nerves, covers, and complexes

Previously, we have already seen how to calculate simplicial homology for topological spaces. We restricted ourselves to those spaces that permit a representation by simplicial complexes. Using this representation, we were able to calculate the Betti numbers, for example, by reducing certain matrices. Real-world data is not commonly endowed with a simplicial structure. As a first step, a point cloud thus needs to be converted into the realm of algebraic topology. We need a certain mathematical rigour here in order to show that the constructions are well-defined and have some expressive power. We shall take a look at a very general concept first, the *nerve of a covering*. From this, we will then derive two related constructions that permit us to approximate data by simplicial complexes.

Definition 4.1 (Covering of a topological space). Given a topological space $\mathbb{X}$, an indexed family of sets $\mathcal{U} := \{U_i \mid i \in \mathcal{I}\}$ is a *cover* of $\mathbb{X}$ if

$$\mathbb{X} \subseteq \bigcup_{i \in \mathcal{I}} U_i, \tag{4.1}$$

i.e. the topological space is contained within the union of sets. In practice, we assume that the number of sets required for a covering is finite. We will later on see how to obtain coverings automatically for data from $\mathbb{R}^n$.



A discrete topological space A covering

Figure 4.1: A topological space and a covering. Each set of the covering is shown as a disk. Some of the points are covered more densely than others.

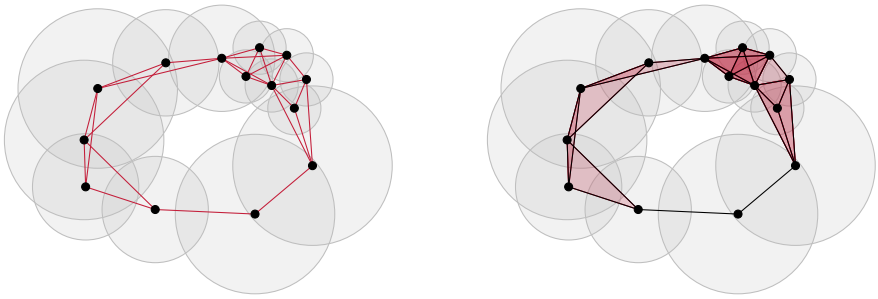
Definition 4.2 (Nerve of a covering). Given a covering $\mathcal{U}$ of open sets of a topological space, the *nerve* of $\mathcal{U}$ consists of all non-empty subsets whose common intersection is non-empty. Formally, we have:

$$\text{Nerve}\,\mathcal{U} := \left\{U_i \subseteq \mathcal{U} \mid \bigcap U_i \neq \varnothing\right\} \tag{4.2}$$

From the previous definition, we see that if a set $\sigma \in \text{Nerve}\,\mathcal{U}$, then for each $\tau \subseteq \sigma$ we have $\tau \in \text{Nerve}\,\mathcal{U}$, as well. Nerve $\mathcal{U}$ thus meets all the requirements of an abstract simplicial complex according to Definition 3.3 on p. 27.

Figure 4.1 illustrates how a covering of a simple discrete topological space may look like. Given a covering of a topological space, we may thus obtain a simplicial complex from its nerve by checking all subsets for common intersections. The 1-skeleton of this *nerve complex* is thus given by all pairs of cover sets that intersect, while the 2-skeleton contains all triples of cover sets that intersect, and so on. Figure 4.2 depicts the 1-skeleton and the 2-skeleton of a covering. For higher-dimensional simplices, this construction becomes more and more cumbersome. In essence, to obtain the complete nerve complex, we need to enumerate all subsets of all cover sets in the covering. This has a complexity of $\mathcal{O}(2^n)$, where n is the cardinality of the covering.

Nonetheless, the nerve of a covering is an interesting and useful construction because it preserves the *homotopy type* of a finite family of sets. This is a formal way of stating that the construction is capable of representing topological objects correctly both in the continuous and in the discrete case. We first require some auxiliary constructions.



The 1-skeleton of the nerve The 2-skeleton of the nerve

Figure 4.2: The nerve of a covering. To find out whether a k-simplex is a member of the nerve, we need to evaluate intersections between subsets of cardinality k. When calculating the different skeletons of the nerve, we can see that the upper right part of the topological space gives rise to many 2-simplices.

Definition 4.3 (Homotopy). Let $\mathbb{X}$ and $\mathbb{Y}$ be topological spaces. Furthermore, for $t \in [0,1]$, let $h_t \colon \mathbb{X} \to \mathbb{Y}$ be a family of maps. This family is called a *homotopy* if the associated map

$$\begin{aligned} H \colon \mathbb{X} \times [0,1] &\to \mathbb{Y} \\ (x,t) &\mapsto h_t(x) \end{aligned} \tag{4.3}$$

is continuous. We may think of a homotopy as deformation that happens over time. Homotopy is a less powerful, but also less constraining, concept than homeomorphism. Two maps $f, g \colon \mathbb{X} \to \mathbb{Y}$ are said to be *homotopic* if there is a homotopy h_t connecting them. We denote this by $f \simeq g$.

We are particularly interested in pairs of functions that are homotopic to the corresponding identity function of a topological space. In essence, these pairs demonstrate that two spaces are equivalent in the sense that they can be deformed into each other.

Definition 4.4 (Homotopy equivalence). A map $f \colon \mathbb{X} \to \mathbb{Y}$ is a *homotopy equivalence* if there is a map $g \colon \mathbb{Y} \to \mathbb{X}$ such that $f \circ g \simeq \mathrm{id}_{\mathbb{Y}}$ and $g \circ f \simeq \mathrm{id}_{\mathbb{X}}$. In other words, a homotopy equivalence is given by two functions whose composition is homotopic to the identity function on both spaces. Again, this notion is less powerful but more generic than the concept of homeomorphic spaces.

Definition 4.5 (Homotopy type). Two topological spaces $\mathbb{X}$ and $\mathbb{Y}$ have the same *homotopy type* if a homotopy equivalence between them exists. This is often denoted by $\mathbb{X} \simeq \mathbb{Y}$.

For example, the space containing a single point $\{*\}$ has the same homotopy time as the real line $\mathbb{R}$. Intuitively, we know that this must be the case because we can always shrink the line down to a point or expand the point into a line. To see this, let us define two

functions $f\colon \{*\} \to \mathbb{R}$ and $g\colon \mathbb{R} \to \{*\}$ as $* \mapsto 0$ and $x \mapsto *$, respectively. We have $g \circ f = \mathrm{id}_{\{*\}}$ by definition. To see that $f \circ g \simeq \mathrm{id}_{\mathbb{R}}$, we need to define a homotopy. Let $g_t\colon \mathbb{R} \to \mathbb{R}$ be defined by $x \mapsto (1-t)x$. Then $g_0 = \mathrm{id}_{\mathbb{R}}$ and $g_1 = f \circ g$. Hence, $f \circ g \simeq \mathrm{id}_{\mathbb{R}}$ and we have proven that both spaces have the same homotopy type.



Having introduced the required concepts, we may now state precisely why the nerve of a covering is a useful construction. In essence, it preserves the *homotopy type* of our data. Hence, it does not introduce additional artefacts during the analysis.

Theorem 4.6 (Nerve theorem). Let $\mathcal{U}$ be a finite family of closed, convex sets in Euclidean space. Then Nerve $\mathcal{U}$ and the union of sets in $\mathcal{U}$ have the same homotopy type.

Proof. The first rigorous proof of this theorem was given by Borsuk [51]. A proof in terms of modern algebraic topology is given by Hatcher [192, pp. 459–460]. ∎

Having established the utility of the nerve complex calculation, we return back to the issue we raised previously. How do we obtain a cover of our data? A simple construction would use closed geometric balls in Euclidean space (with the same radius each). This idea leads us to the Cěch complex.

Definition 4.7 (Čech complex). Letting P be a finite set of points in $\mathbb{R}^n$, and $\mathrm{B}_x(r)$ denote a ball with centre $x \in \mathbb{R}^n$ and radius $r \in \mathbb{R}$, the *Čech complex* of P and r is the nerve of this family of balls:

$$\mathcal{C}(r) := \left\{ \sigma \subseteq P \,\middle|\, \bigcap_{x \in \sigma} \mathrm{B}_x(r) \neq \varnothing \right\} \tag{4.4}$$

The previous definition bears a close similarity to Definition 4.2, which defines the nerve of a covering. The Čech complex may be seen as the Euclidean equivalent of the nerve of a covering for a specific family of covers—namely the ones defined by Euclidean balls.

The Čech complex has the nice property that increasing radii result in nested complexes. Thus, if $r \leq r'$, we have $\mathcal{C}(r) \subseteq \mathcal{C}(r')$. This fact shall prove to be very relevant later on. For a radius r, a subset of vertices σ forms a simplex if and only if the corresponding set of points can be enclosed within a ball of radius r [141, p. 72]. Checking whether a set of points can be enclosed by a ball of a given radius is a standard problem in computational geometry [165, 175]. Its solution becomes more complex with increasing dimensions, making the Čech complex currently infeasible to calculate for most applications. There have been recent attempts at an improved construction algorithm [121], but the results are still somewhat preliminary.

Thus, while we would prefer being able to work with the Čech complex, we have to resort to another type of complex that is more computationally tractable. The drawback of the subsequent construction is that we lose the homotopy type preservation property to some extent. However, the combinatorial construction turns out to be less complicated.

Definition 4.8 (Diameter). Let $\mathbb{X}$ be a finite topological space with an associated metric $\operatorname{dist}_{\mathbb{X}}$. The *diameter* of $\mathbb{X}$ is the upper bound of the set of all pairwise distances, i.e.

$$\operatorname{diam} \mathbb{X} := \sup\{\operatorname{dist}_{\mathbb{X}}(x, y) \mid x, y \in \mathbb{X}\}, \tag{4.5}$$

and since $\mathbb{X}$ is finite, the supremum always exists and is attained by some pair of points. The definition extends to subsets in a natural manner.

Definition 4.9 (Vietoris–Rips complex). Given a scale parameter r and a finite set of points P, the Vietoris–Rips complex is defined as the simplicial complex that contains all subsets whose diameter is at most r:

$$\mathcal{V}(r) := \{\sigma \subseteq P \mid \operatorname{diam} \sigma \leq r\} \tag{4.6}$$

The complex thus contains a simplex $\sigma = \{v_0, \ldots, v_k\}$ if and only if all points are within a distance of at most r to each other. This construction is due to Vietoris [375] and has nowadays become the standard way of generating simplicial complexes from point clouds.

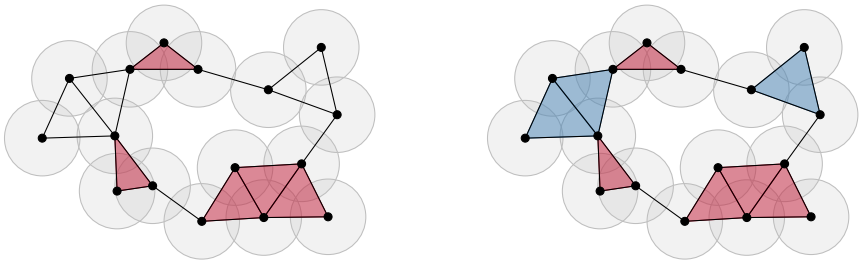
There is no equivalent to Theorem 4.6 for the Vietoris–Rips complex. How can we ascertain the fidelity of the topological approximation? We first observe that both complexes are related.

Lemma 4.10. For a given radius r, we have the following nesting relation:

$$\mathcal{C}\left(\frac{1}{2}r\right) \subseteq \mathcal{V}(r) \subseteq \mathcal{C}(r) \tag{4.7}$$

Proof. We first show that $\mathcal{C}\left(\frac{1}{2}r\right) \subseteq \mathcal{V}(r)$. Let $\sigma \in \mathcal{C}\left(\frac{1}{2}r\right)$. By Definition 4.7, we know that there is a ball with radius $\frac{1}{2}\epsilon$ that encloses all points of σ. The diameter of such a ball is at most ϵ. Hence, $\sigma \in \mathcal{V}(r)$. To show the other inclusion, let $\tau \in \mathcal{V}(r)$. By Definition 4.9, $\operatorname{diam} \tau \leq r$. A ball with radius r, centred at any of the points in τ is guaranteed to contain the points in τ. Thus, $\tau \in \mathcal{C}(r)$ and the claim follows. ∎

The Vietoris–Rips complex is an example of a *flag complex*, or a *clique complex*, which means that its simplices are completely determined by its 1-skeleton, i.e. its vertices and edges. As a short-hand notation, we will write $\mathcal{C}_\epsilon$ and $\mathcal{V}_\epsilon$ to refer to the Čech complex and the



Čech complex $\mathcal{C}_{\frac{1}{2}\epsilon}$ Vietoris–Rips complex $\mathcal{V}_\epsilon$

Figure 4.3: An example showing how to obtain geometric complexes from a simple data set. Both complexes are shown for the same value of the scale parameter. The Čech complex contains a triangle for each subset of three balls with a non-empty intersection. By contrast, the Vietoris–Rips complex contains a triangle whenever three balls have a pairwise intersection.

Vietoris–Rips complex of scale ϵ. Figure 4.3 depicts the construction of the Čech complex and the Vietoris–Rips complex. The new 2-simplices that are part of $\mathcal{V}_\epsilon$ but not part of $\mathcal{C}_{\frac{1}{2}\epsilon}$ have been highlighted.

Depending on the application, other constructions for creating simplicial complexes are available as well. The *flow complex* [179] is often used to reconstruct two-dimensional surfaces. It is closely related to the *Delaunay triangulation* [166]. Since these triangulations tend to become very large, Edelsbrunner et al. [147] introduced *alpha complexes*, which are subcomplexes of the Delaunay triangulation and permit users to tune the scale parameters that are used for constructing the complex. In a subsequent publication, Edelsbrunner and Mücke [152] extended the algorithm to work in 3D. In practice, all of these complexes do not scale very well beyond three dimensions, making them unsuitable for the data sets used in this thesis.

4.2 Calculating the Vietoris–Rips complex

In the previous definitions, we have completely ignored how the complexes are generated. When we talk about intersections and diameters, this presupposes the existence of some *distance measure* such as the common Euclidean distance. One of the strengths of the complexes defined above is that they work for a large family of distance measures—e.g. the ones used by domain experts to analyse their data. In the following, we will assume that we have a function $\mathrm{dist}(\cdot,\cdot)$ for measuring distances between individual data points. If not specifically mentioned, we do not require this function to be a metric in the mathematical sense.

Prior to calculating any sort of flag complex such as the Vietoris–Rips complex, an algorithm requires a way to define neighbourhood relations. This is often solved by calculating a *Rips graph* (also known as a *neighbourhood graph*).

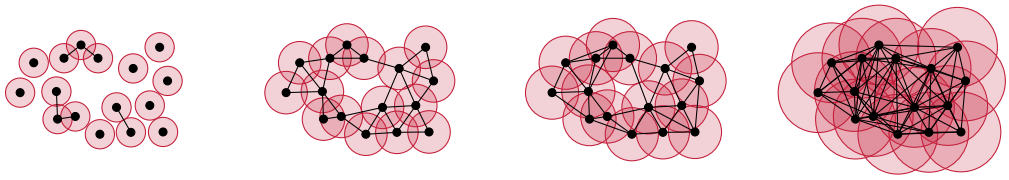


Figure 4.4: The effects of varying ϵ for the Rips graph construction. A small value for ϵ results in a very sparse Rips graph $\mathcal{R}_\epsilon$. As ϵ increases, more and more points become neighbours.

Definition 4.11 (Rips graph). The Rips graph at scale ϵ of a set of points $P = \{p_1, p_2, \dots\}$, with $p_i \in \mathbb{R}^d$, contains all edges between points that are within a distance of less than or equal to ϵ to each other. Formally, $\mathcal{R}_\epsilon = (V, E)$ is a graph with a set of vertices V and a set of edges E:

$$V := \{1, 2, \dots\} \tag{4.8}$$

$$E := \big\{(u, v) \mid \operatorname{dist}(p_u, p_v) \leq \epsilon\big\} \tag{4.9}$$

The distance measure $\operatorname{dist}(\cdot,\cdot)$ is not required to be a metric in the mathematical sense, making this construction very general.

We have already seen that the Rips graph is the 1-skeleton of the corresponding Vietoris–Rips complex. Its shape depends on the values of the ϵ parameter. At the extremal ends of the spectrum, the graph may degenerate into a disconnected set of points (no edges) or the complete graph K_n of the n input points (all possible edges). Figure 4.4 illustrates how the Rips graph changes when increasing ϵ. To include additional information about the scale of the Rips graph, edges are usually being assigned their distance values as a weight. We may thus visualize the Rips graph of a given data set as being constructed by adding edges with increasing distance, until a certain 'resolution' has been reached. In Chapter 5, Section 5.4, p. 96 ff., we will develop algorithms for choosing suitable values for ϵ automatically.

Calculating a Rips graph

The construction of the Rips graph entails the calculation of nearest neighbours in high-dimensional spaces. This problem cannot be solved efficiently—either space or time requirements tend to grow exponentially with the dimension [208]. However, if we are satisfied with *approximate nearest neighbours*, it turns out that we can calculate the Rips graph very efficiently in higher dimensions. Using k-d trees [36], approximate solutions may be obtained with an expected complexity of $\mathcal{O}\big(\log n\, \epsilon^{-d}\big)$ for the query time, with $\mathcal{O}(n)$ space and $\mathcal{O}(n \log n)$ construction time [12]. Hence, the Rips graph is an efficient and scalable construction.

In practice, we use the *fast library for approximate nearest neighbours* (FLANN) [275] to implement the Rips graph construction. This C++ library supports different kinds of Minkowski metrics, such as the Manhattan distance, the Euclidean distance, and the Hamming distance.

Calculating a Vietoris–Rips complex

To calculate the Vietoris–Rips complex from a Rips graph, we use a fast and efficient algorithm by Zomorodian [407]. The algorithm calculates the k-skeleton of $\mathcal{V}_\epsilon$, i.e. all simplices up to dimension k. The main idea is to add new vertices incrementally and construct all of their cofaces for which they are maximal, i.e. for which there are no proper cofaces. See Algorithm 1 for a pseudo-code description of the expansion.

Algorithm 1: Incremental expansion of the Vietoris–Rips complex

```
function IncrementalExpansion(R_ε, d)
    V_ε ← ∅
    for Vertex v ∈ R_ε do
        N ← LowerNeighbours(R_ε, v)
        AddCofaces(R_ε, d, {v}, N, V_ε)
    end for
    return Vε
end function

function AddCofaces(R_ε, d, σ, N, V_ε)
    V_ε ← V_ε ∪ {σ}
    if |σ| ≥ d then
        return
    end if
    for Vertex v ∈ N do
        τ ← σ ∪ {v}
        M ← N ∩ LowerNeighbours(R_ε, v)
        AddCofaces(R_ε, d, τ, M, V_ε)
    end for
end function

function LowerNeigbours(G, u)
    N ← ∅
    for Vertex v ∈ G do
        if v < u and {u, v} ∈ G then
            N ← N ∪ v
        end if
    end for
end function
```

For each vertex, the algorithm calls the function LowerNeighbours to enumerate all neighbours with a smaller index. This ensures that simplices are neither visited nor added multiple times while building the complex. Using the list of lower neighbours, the algorithm

now adds all cofaces of increasing dimensionality for which the current vertex is maximal. To this end, the algorithm traverses the list of neighbours and extends the given simplex by one vertex. This is valid because recalling Definition 4.9, we only require points to have pairwise intersections in order to form a simplex.

Weights

The Rips graph $\mathcal{R}_\epsilon$ at scale ϵ affords a natural set of weights, given by the distance function $\operatorname{dist}(\cdot,\cdot)$ used to calculate it. We assign each edge (u, v) the weight $\operatorname{dist}(p_u, p_v)$. These weights can be extended to the Vietoris–Rips complex $\mathcal{V}_\epsilon$.

Definition 4.12 (Weight function). Let $\mathcal{V}_\epsilon$ be a Vietoris–Rips complex with a corresponding Rips graph. We have a natural *weight function* $w\colon \mathcal{V}_\epsilon \to \mathbb{R}$ for each simplex:

$$w(\sigma) := \begin{cases} 0 & \text{if } \sigma \text{ is a vertex} \\ d(p_u, p_v) & \text{if } \sigma = \{u, v\} \\ \max_{\tau \subseteq \sigma} w(\tau) & \text{else} \end{cases} \tag{4.10}$$

This weight function stores the minimum value for ϵ at which a given simplex 'enters' the Vietoris–Rips complex $\mathcal{V}_\epsilon$. Taking the maximum ensures that the complex can be sorted consistently. This will become relevant when calculating persistent homology.

Performance aspects

Although the Vietoris–Rips complex is computationally tractable for most data, it does not scale well. In the worst case, $\mathcal{V}_\epsilon$ may grow to $\mathcal{O}(2^n)$ simplices, where n is the cardinality of the input data. Recent work in computational geometry thus concentrates on mitigating this issue. Zomorodian [408], for example, introduced *tidy sets* that permit the faster calculation of simplicial homology in clique complexes. The approach does not extend to persistent homology, though. Sheehy [334] provides a linear-size approximation to the Vietoris–Rips complex that results in smaller complexes. Buchet et al. [61] extend this approximation to the case of complexes with arbitrary weights and provide an efficient implementation. Cavanna et al. [82] take a more geometrical perspective and prove that vertex removal—which is required to obtain smaller complexes—may be implemented as a sequence of elementary edge collapses. All of these approaches make heavy use of the *net-tree* data structure developed by Har-Peled and Mendel [188], which permits a hierarchical description of scale information in data.

A different approach employs subsampling techniques. For extremely large data sets, subsampling is based on the reasonable assumption that large-scale behaviour may be extracted

even from smaller sets of samples. Following this idea, de Silva and Carlsson [337] introduced *witness complexes*. By using a smaller, representative subset of samples from the data, witness complexes remain computationally tractable. Statistically, witness complexes are capable of extracting large-scale features with high confidence. Guibas and Oudot [183] proved that witness complexes and restricted Delaunay triangulations are closely related in 2D and 3D. This result puts witness complexes on sure topological footing as it guarantees that manifolds can be properly reconstructed. Unfortunately, Boissonnat et al. [48] show that these properties do not necessarily hold in higher-dimensional spaces.

Expressive power

By Lemma 4.10, we already know that the Vietoris–Rips complex has the potential to be a useful approximation to the Čech complex. It is possible to obtain tighter bounds for certain classes of spaces. Latschev [233] proves that Vietoris–Rips complexes are capable of reconstructing Riemannian manifolds correctly, provided they are built using samples that are close to the manifold with respect to the Gromov–Hausdorff distance, which we will encounter in Theorem 4.31 on p. 75. Attali and Lieutier [14] extend this result and show that homotopy type preservation is possible if the L_∞-distance is used. In a more recent publication, Attali et al. [16] show that Vietoris–Rips complexes using the Euclidean distance have similar preservation properties, provided some mild conditions pertaining to the convexity of the given space hold. The Vietoris–Rips complex is thus a useful approximation of the intrinsic geometry and topology of a data set. Later on, we shall also discuss some results for quantifying the stability of the topological approximation in terms of persistent homology.

4.3 Calculating 0-dimensional persistent homology

Before formally introducing and defining persistent homology in Section 4.4, we shall first look at 0-dimensional persistent homology to motivate the subsequent ideas. We start with a simple question, namely how to describe the topology of a function $f\colon \mathbb{D} \subseteq \mathbb{R} \to \mathbb{R}$. What are the topological features that we may expect such a function to have? Following the idea of *Morse theory* [270, 273], we assume that we are interested in connectivity changes of the sublevel sets of f, as described by Definition 2.2 on p. 19. The algorithm we will develop in this section works analogously for the superlevel sets—only the traversal order will have to be reversed. In homology terms, this is akin to analysing the zeroth Betti number β_0 of the function. Of course, *eventually* we will have $\beta_0 = 1$ if we assume the function to be connected. We shall see, however, that the changes in connectivity convey a large amount of interesting information about the multi-scale behaviour of the function.

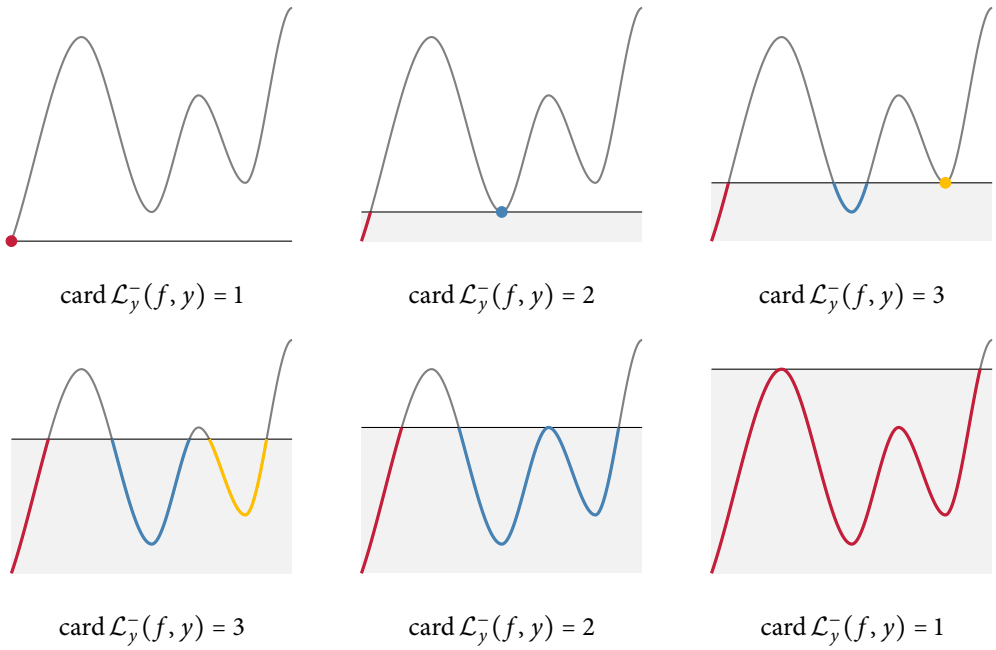


Figure 4.5: The 0-dimensional persistent homology of a function $f\colon \mathbb{D} \subseteq \mathbb{R} \to \mathbb{R}$ can be calculated by traversing the function values from lowest to highest, stopping at local extrema. The caption of every figure displays the number of connected components in the sublevel set of the function. We shall see later on that this traversal corresponds to a *sublevel set filtration*.

In the following, we will use Figure 4.5 as an illustration. Starting from the lowest function value, we traverse the function values, using a *union–find* data structure [113, Chapter 21] to keep track of the connected components in the current sublevel set of the function. We observe that the number of connected components changes only when the traversal arrives at a local extremum. A local minimum creates a new connected component. This component then continues to grow until we reach a local maximum. Here, two connected components meet and merge into one. For consistency reasons, we merge the 'younger' component, i.e. the one corresponding to the larger function value, into the 'older' component, i.e. the one corresponding to the lower function value. Using this terminology, the first connected component is the 'oldest' because it is created by the global minimum. Algorithm 2 gives a high-level description of this process. At the end of the traversal, only one connected component is left. This component can never merge with another one.

Going beyond simplicial homology

We have already claimed that this traversal is equivalent to calculating the zeroth Betti number β_0. At the end of Algorithm 2, we have $\beta_0 = 1$, which is indeed correct because f only

has a single connected component. However, we may obtain more information during the traversal. Whenever we perform a merge between two connected components, we can make note of the function values. Supposing we merge two connected components with function values $c' \leq c$ at a local maximum with function value d. In this case, we store the tuple (c, d). It signifies that a connected component that appeared in the sublevel set for $f(x) = c$ merges with another connected component in the sublevel set for $f(x) = d$. The tuples thus pair related minima and maxima with each other, yielding a multi-scale description of the function. This process bears a close resemblance to the theory of *size functions* [42, 172]. We shall see later on how to use this information as a shape descriptor with well-defined metrics.

Algorithm 2: Calculating 0-dimensional persistent homology

Require: A function $f\colon \mathbb{D} \subseteq \mathbb{R} \to \mathbb{R}$

```
1:  function PersistentHomology(f)
2:      U ← ∅                                  ▷ Initialize an empty union–find structure
3:      Sort the function values of f in ascending order.
4:      for Function value y of f do
5:          if y is a local minimum then
6:              Create a new connected component in U.
7:          else if y is a local maximum then
8:              Use U to merge the two connected components meeting at y.
9:          else
10:             Use U to add y to the current connected component.
11:         end if
12:     end for
13: end function
```

Connection to discrete functions & Rips graphs

We have seen how to calculate 0-dimensional persistent homology for an idealized function. In practice, we are given a set of discrete samples of a function. Algorithm 3 gives an in-depth description of the persistent homology calculation for this case. It is only slightly more complicated than the idealized case.

The concepts used in both algorithms extend by *functoriality* [232, Chapter I] onto all objects that permit a connectivity description, such as Rips graphs. Instead of the implicit connectivity given by the index of a function value, we now need to query the graph for the existence or non-existence of edges. Sorting the vertices and edges by increasing weight, we can keep track of connected components in the graph in a similar manner than for functions. The tuples generated by this procedure describe the geometrical–topological behaviour of f on the Rips graph. A similar construction is possible for networks or arbitrary graphs. As long as we are able to define a traversal order for the vertices and edges of the graph, we can calculate its 0-dimensional persistent homology.

Algorithm 3: Calculating discrete 0-dimensional persistent homology

Require: A discrete sample $\{(x_1, y_1), (x_2, y_2), \dots\}$ of a function $f\colon \mathbb{D} \subseteq \mathbb{R} \to \mathbb{R}$

function PersistentHomology(f)
  U $\leftarrow \varnothing$ ▷ Initialize an empty union–find structure
  Sort the value tuples in ascending order, such that $y_1 \geq y_2 \geq \dots$
  for Tuple (x_i, y_i) of f **do**
    if $y_{i-1} > y_i$ and $y_{i+1} > y_i$ **then** ▷ y_i is a local minimum
      U.add(i) ▷ Create a new connected component in U
    else if $y_{i-1} < y_i$ and $y_{i+1} < y_i$ **then** ▷ y_i is a local maximum
      $c \leftarrow$ U.get$(i-1)$ ▷ Get first connected component
      $d \leftarrow$ U.get$(i+1)$ ▷ Get second connected component
      U.merge(c, d) ▷ Merge the two connected components meeting at y_i
    else ▷ y_i is a regular point
      $c \leftarrow$ U.get$(i-1)$ ▷ Get connected component
      U$[c] \leftarrow$ U$[c] \cup i$ ▷ Add y_i to the current connected component
    end if
  end for
  return U
end function

Performance

Calculating 0-dimensional persistent homology is highly efficient and scalable. The algorithm requires a single pass through the object. For a function, the number of visited points is of the order of $\mathcal{O}(n)$, where n denotes the number of function values. In case of a Rips graph, the algorithm requires us to visit all vertices and all edges, which are of the order of $\mathcal{O}(|V| + |E|)$ or, rather pessimistically, $\mathcal{O}(n^2)$, where n is the number of vertices. During each iteration, a result of Tarjan [355] tells us that every operation with the union–find data structure has an amortized complexity of $\mathcal{O}(\alpha(n))$, where n is the cardinality of the input data and $\alpha(\cdot)$ denotes the extremely slow-growing inverse of the Ackermann function. We have $\alpha(n) < 5$ for all practical values of n. Hence, the amortized runtime of 0-dimensional persistent homology calculations remains linear in the cardinality of the object.

Connection to single-linkage dendrograms

Before deriving an algorithm to calculate persistent homology in the general case, we first want to explain some similarities between the previous algorithms and the well-known *single-linkage clustering algorithm* [304]. A central component of the previous calculation was to keep track of how the connected components of the sublevel sets changed. We described changes by means of tuples (c, d), where $c \in \mathbb{R}$ and $d \in \mathbb{R}_\infty$. We may connect this information to the *dendrogram* of single-linkage clustering in a natural manner. First, we consider every data point to be a vertex in a tree. When merging two connected components, we then

insert an edge between the two corresponding vertices. This process yields a tree—or a forest in case multiple connected components remain—that is exactly the dendrogram we would obtain when calculating a single-linkage clustering on the data. Persistent homology may thus be seen as a higher-dimensional analogue to clustering analysis, or as a *forgetful functor* in the sense of category theory [232, p. 14]. In contrast to clustering algorithms, it focuses on higher-dimensional connectivity information. Recent work by Carlsson and Mémoli [68, 69] focuses on developing frameworks for describing hierarchical clustering methods in terms of topological properties.

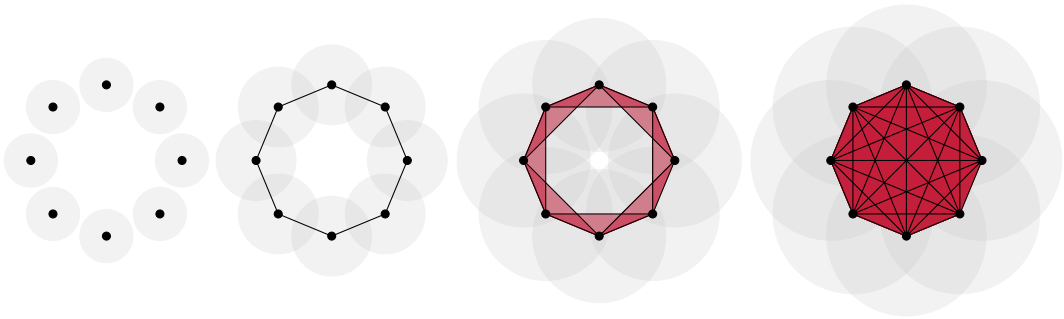
4.4 Calculating persistent homology

We have now seen how to calculate 0-dimensional persistent homology for discrete functions and, by functoriality, on all objects that admit connectivity relations. In the most general case, we want to calculate persistent homology for a Vietoris–Rips complex $\mathcal{V}_\epsilon$ (or any other simplicial complex) of a data set. Since $\mathcal{V}_\epsilon$ is a simplicial complex, we could conceivably calculate its Betti numbers (using the standard Smith normal form decomposition) and use said information as a shape descriptor of the input data. Unfortunately, it turns out that Betti numbers are extremely volatile. The addition of a single simplex may already change the Betti numbers significantly. Figure 4.6 illustrates this for points arranged in a circle. As soon as a critical threshold for ϵ has been reached, the Vietoris–Rips complex does not contain a hole any more, meaning that the circle will be treated as a closed object without any topological features. This instability is not just a result of the selected geometric complex; the Čech complex and all other complexes suffer from the same deficits—if an unsuitable threshold has been selected, the Betti numbers of the complex may be inaccurate. How can we make the calculation of Betti numbers more stable? Edelsbrunner et al. [141, 142, 148, 149] observed that the calculations become more robust if we endow our simplicial complex with a function f. Instead of considering all simplices 'at the same time', we can use f to define a sort of 'scale' for defining the complex. To this end, we first require some additional definitions.

Definition 4.13 (Monotonic function). Let K be a simplicial complex and $f\colon \mathrm{K} \to \mathbb{R}$ be a real-valued function defined on the complex. f is *monotonic* if its value does not decrease when going to the cofaces of a simplex. Formally, we require

$$f(\sigma) \leq f(\tau) \tag{4.11}$$

for every simplex $\sigma \subseteq \tau$. If f satisfies this condition, we will also say that f is *compatible* with the simplicial complex K.



(a) $\epsilon = 0.35$ (b) $\epsilon = 0.53$ (c) $\epsilon = 0.88$ (d) $\epsilon = 1.05$

Figure 4.6: The instability of Betti numbers. We calculate Vietoris–Rips complexes at different scales for eight points arranged on a unit circle. What is the 'correct' value for ϵ? Focusing on the first Betti number β_1, we have $\beta_1 = 1$ only for the complexes (b) and (c). As long as $\epsilon \leq 0.50$, we have $\beta_1 = 0$, as depicted by (a). After $\epsilon \geq 1.0$, on the other hand, we also have $\beta_1 = 0$ because the hole of the circle has been closed and, as shown in (d), all available simplices have been added. The key idea of persistent homology is to realize that for a *range* of ϵ, namely for $\epsilon \in [0.50, 1.0)$, we have $\beta_1 = 1$. The hole thus *persists* over this scale.

Definition 4.14 (Filtration). For a monotonic function on a simplicial complex, the sublevel sets of the function f are simplicial subcomplexes. We obtain a *filtration* if we arrange them as an increasing sequence, i.e.

$$\varnothing = \mathrm{K}_0 \subseteq \mathrm{K}_1 \subseteq \cdots \subseteq \mathrm{K}_{n-1} \subseteq \mathrm{K}_n = \mathrm{K}, \tag{4.12}$$

where each K_i may be thought of as being created by a unique function value y_i of f. We do not require this restriction in practice, though. A filtration may be obtained from a weighted simplicial complex, such as the Vietoris–Rips complex, provided simplex weights have been assigned in a compatible way. If a simplex σ has a weight w, all of the cofaces of σ need to have a weight that is at least as large as w. In this case, we may sort simplices by their weights and, in case of ties, lexicographically, in order to obtain a valid filtration.

✦

Filtrations occur naturally when building geometric complexes. The Vietoris–Rips complex, for example, with its weight function given by Definition 4.12, automatically results in a valid filtration. The nesting relation of simplicial complexes in a filtration induces an inclusion map, which in turn induces an homomorphism between the corresponding homology groups. We denote this homomorphism by

$$f_p^{i,j} \colon H_p(\mathrm{K}_i) \to H_p(\mathrm{K}_j), \tag{4.13}$$

where p is the dimension and we have $i \leq j$ to ensure a proper ordering. Hence, a filtration gives rise to a sequence of homology groups that are connected via the functions $f_p^{i,j}$, i.e.

$$0 = H_p(\mathrm{K}_0) \xrightarrow{f_p^{0,1}} H_p(\mathrm{K}_1) \xrightarrow{f_p^{1,2}} \dots \xrightarrow{f_p^{n-2,n-1}} H_p(\mathrm{K}_{n-1}) \xrightarrow{f_p^{n-1,n}} H_p(\mathrm{K}_n) = H_p(\mathrm{K}), \tag{4.14}$$

where p again denotes the dimension of the homology groups. As the filtration increases in size, the homology groups change—some homology classes vanish, some homology classes are created, and so on. Since the filtration induces an ordering, we can collect homology classes according to the threshold (with respect to the filtration) at which they appear or disappear. Following Definition 3.13 on p. 32 and Definition 3.14 on p. 33, the induced homomorphisms permit us to define persistent homology groups.

Definition 4.15 (Persistent homology group). Given two indices $i \leq j$, the p^{th} persistent homology group $H_p^{i,j}$ is defined as

$$H_p^{i,j} := Z_p(\mathrm{K}_i) / \big(B_p\big(\mathrm{K}_j\big) \cap Z_p(\mathrm{K}_i)\big), \tag{4.15}$$

i.e. $H_p^{i,j}$ contains all the homology classes of K_i that are still present in K_j. Following the notation from above, we have $H_p^{i,i} = H_p(\mathrm{K}_i)$.

We can describe classes in different persistent homology groups in a more precise manner. We say that a class $c \in H_p(\mathrm{K}_i)$ is *created* in K_i if $c \notin H_p^{i-1,i}$. Hence, c did not appear in any of the previous persistent homology groups. Similarly, for c created in K_i, we say that it is *destroyed* in K_j, with $i \leq j$, if the class merges with an older class when traversing the filtration from K_{j-1} to K_j. Formally, this means that $f_p^{i,j-1}(c) \notin H_p^{i-1,j-1}$, and $f_p^{i,j}(c) \in H_p^{i-1,j}$. This means that the class is not part of any earlier persistent homology group but gets merged into the image of one by the inclusion between K_i and K_j. Such a class c thus *persists* from K_i to K_j. We can express this more formally.

Definition 4.16 (Persistence). Let c be a homology class that gets created in K_i and destroyed in K_j. Furthermore, let y_i and y_j be the corresponding function values. We then say that the class c has a *persistence* of

$$\operatorname{pers} c := y_j - y_i, \tag{4.16}$$

with $\operatorname{pers} \in \mathbb{R}$ because both function values are assumed to be real. If a homology class c is never destroyed by the given filtration, we set $\operatorname{pers} c := \infty$ and refer to it as an *essential homology class*. To account for essential homology classes, we permit $\operatorname{pers}(\cdot)$ to take values in $\mathbb{R}_\infty := \mathbb{R} \cup \{\infty\}$, the set of extended real numbers.

Using persistence The persistence of a homology class serves as relevance criterion. It is often used for simplifying functions on manifolds [148, 150] or 'pruning away' undesired features in scalar fields [186, 362]. A low persistence value indicates a small-scale feature. As noise is commonly taken to be a small-scale phenomenon, topological features with low persistence values are often considered noise.

Persistence pairs By tracking each homology class through the filtration and recording creation as well as destruction events, we obtain a set of tuples that describe how the homology of the input simplicial complex changes as the threshold is changed. Each of these *persistence tuples* or *persistence pairs* is of the form (c, d) with $c \in \mathbb{R}$ and $d \in \mathbb{R}_\infty$. c indicates the threshold at which a homology class was created, while d indicates the threshold at which it was destroyed. In Section 4.5, we will see how to visualize these tuples. The definition of persistent homology groups also permits a generalization of the Betti numbers.

Definition 4.17 (Persistent Betti number). Given two indices $i \leq j$, the p^{th} persistent Betti number is defined as

$$\beta_p^{i,j} := \operatorname{rank} H_p^{i,j}, \tag{4.17}$$

which is completely analogous to the definition of Betti numbers for simplicial homology. Persistent Betti numbers are often used to quantify *stability regions* in data.

So far, the definition of persistent homology groups may look rather daunting. We have to track every homology class through all possible complexes, after all. However, using the *nesting property* of the Vietoris–Rips complex, we obtain an efficient algorithm that requires the reduction of a single matrix in order to obtain all persistence tuples in all dimensions. This requires a consistent ordering of simplices based on their weights. We thus sort all simplices by their weights, giving lower-dimensional simplices precedence over higher-dimensional simplices, and breaking ties arbitrarily. This yields a full ordering $\sigma_1, \sigma_2, \ldots$ of the simplices of the Vietoris–Rips complex $\mathcal{V}_\epsilon$. We then create a large boundary matrix ∂ that contains all simplices. The matrix is defined by

$$\partial_{ij} = \begin{cases} 1 & \text{if } \sigma_i \text{ is part of the boundary of } \sigma_j \\ 0 & \text{else} \end{cases}, \tag{4.18}$$

where we only count simplices that are a 'proper face' of another simplex. Hence, for a p-simplex σ, we only consider a $(p-1)$-simplex τ if $\tau \in \partial_p \sigma$. For further calculations, let $\operatorname{low}(j)$ be the row index of the lowest '1' in column j. $\operatorname{low}(j)$ is undefined if the j^{th} column is zero. We call a matrix *reduced* if $\operatorname{low}(j) \neq \operatorname{low}(j')$ for $j \neq j'$, provided both columns are non-zero. Similar to the calculation of the SNF, we can reduce the boundary matrix ∂ by elementary

column operations only. Algorithm 4 describes the reduction procedure. We merely add columns from left to right, which can be implemented very efficiently. Later on, we will describe an algorithm that is more complex but contains a detailed description of topological features in terms of simplicial chains.

Algorithm 4: Reducing the boundary matrix ∂

Require: Boundary matrix ∂
1: **for** Each column j **do**
2: **while** There is another column $j' < j$ with $\text{low}(j') = \text{low}(j)$ **do**
3: Add column j' to column j.
4: **end while**
5: **end for**

From the reduced boundary matrix, we may read off essentially the same information as for simplicial homology. For example, considering only the columns that correspond to p-simplices, the number of zero columns is the rank of the cycle group Z_p, while the number of non-zero columns is the rank of the boundary group B_p. Following Equation 3.12 on p. 37, their difference is thus the p^{th} Betti number β_p of $\mathcal{V}_\epsilon$. At this point, we have gained no more information than by merely calculating Betti numbers. The matrix calculated by the reduction algorithm is certainly not unique, but it turns out that the lowest 1s in each column are. We have the following lemma.

Lemma 4.18 (Pairing lemma). The pairing between rows and columns, induced by $\text{low}(\cdot)$, in the reduced boundary matrix ∂ is *unique* and does not depend on ∂.

Proof. This was alluded to in the seminal paper by Edelsbrunner et al. [148], although the paper does not contain a direct formulation of the proof for arbitrary dimensions. In his thesis, Morozov [272, pp. 41–42] proves this as a part of investigating the effects of swapping simplices in a filtration. Edelsbrunner and Harer [141, pp. 153–154] give a self-contained proof. ■

✦

Having seen that the lowest 1s in the reduced matrix are unique, we can use the terminology of simplicial homology to classify them. Let ∂' be the reduced matrix. If column j of ∂' is zero, its addition to the simplicial complex creates a new homology class. We thus call σ_j a *positive* simplex. If, on the other hand, $\text{low}(j) = i$, the addition of simplex σ_j destroys a homology class created by the simplex σ_i, and we refer to σ_j as a *negative* simplex. Assigning the paired simplices their corresponding function values, we obtain the persistence tuples that describe how long certain topological features persist in the simplicial complex. Note that a simplex can either be positive or negative. There are no simplices that create and destroy a

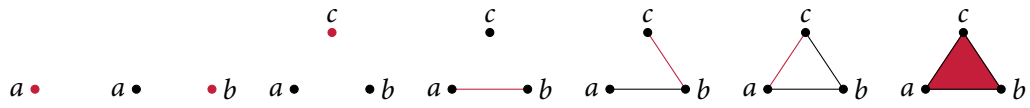


Figure 4.7: An example filtration. The single simplex that is added in each filtration step is highlighted. Since each individual complex must be a valid simplicial complex on its own, cofaces need to precede their faces.

feature simultaneously. We shall subsequently derive an algorithm that exploits this fact by pairing positive and negative simplices. Prior to that, we briefly cover the manual calculation of persistent homology for a simple filtration. This will help build the required intuition and show how to work with the reduced boundary matrix.

Suppose we want to calculate the pairing of simplices according to persistent homology for the filtration shown by Figure 4.7. To obtain a valid filtration according to Definition 4.14, we need to add the vertices, then the edges, followed by the triangle. The boundary matrix is

$$\partial = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{4.19}$$

where the order of columns corresponds to the insertion order. After reduction, we obtain

$$\partial' = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{4.20}$$

from which we read off that simplices $\sigma_4 = \{a, b\}$, $\sigma_5 = \{b, c\}$, and $\sigma_7 = \{a, b, c\}$ are *negative*. The remaining simplices are *positive*. Table 4.1 describes the resulting pairing and briefly explains the reasons for pairing certain simplices with each other.

Creator	Destroyer	Reason
$\{a\}$		The single connected component is never destroyed
$\{b\}$	$\{a, b\}$	The connected component created by b merges with a
$\{c\}$	$\{b, c\}$	The connected component created by c merges with a
$\{a, c\}$	$\{a, b, c\}$	The hole created by the edges is closed by the triangle

Table 4.1: An example pairing. The table above shows the pairing that results from calculating persistent homology for the filtration shown in Figure 4.7. This information may be read off from the reduced boundary matrix in Equation 4.20.

An improved algorithm

The preceding example demonstrated how to obtain a reduced boundary matrix and interpret the resulting pairing properly. What is still missing from the algorithmic perspective is a description of the topological features—the holes—in terms of the simplices of the input complex. To this end, we use an algorithm by Zomorodian and Carlsson [410], which is an improvement of an earlier algorithm [409].

The algorithm rephrases the boundary matrix reductions in terms of 'partners' and 'cascades'. A partner refers to the pairing that occurs during the reduction. If two simplices σ and τ are *partners*, one of them is positive, the other one is negative. This corresponds directly to the relationship established by the low$(\cdot)$ function. The *cascade* of a simplex is a simplicial chain that describes the boundary of the hole created by a positive simplex. It serves as a representative of the homology class created by the simplex. Hence, only positive simplices are assigned a non-empty cascade. Algorithm 5 shows a pseudo-code implementation for the persistent homology calculation.

On a high level, the procedure PairSimplices partitions the filtration into positive and negative simplices. To this end, the function performs the equivalent to column addition in the matrix reduction case. If the boundary of the calculated cascade is empty, the simplex is positive and its cascade is a new cycle. Else, the simplex is negative and destroys a hole—hence, we pair it with the 'youngest' simplex in the filtration, i.e. the one with the largest weight. This is akin to using the low$(\cdot)$ function for selecting a representative simplex. The function EliminateBoundaries is responsible for adding columns—i.e. cascades—to each other. In each iteration, it looks at the youngest simplex τ in the boundary of the cascade, i.e. in the boundary of the modified column. If τ has no partner, i.e. its low$(\cdot)$ value is undefined, the reduction of the current cascade can stop. Else, the hole created by τ has already been destroyed by its partner. We thus add the cascade of its partner to the current cascade. This is equivalent to adding two columns for the matrix reduction algorithm.

Algorithm 5: Persistent homology calculation

```
1:  function PairSimplices(K)
2:      for Simplex σ of the simplicial complex K do
3:          partner[σ] ← ∅
4:          cascade[σ] ← σ
5:          EliminateBoundaries(σ)
6:          if ∂cascade[σ] ≠ ∅ then
7:              τ ← Youngest(∂cascade[σ])
8:              partner[σ] ← τ
9:              partner[τ] ← σ
10:         end if
11:     end for
12: end function

13: function EliminateBoundaries(σ)
14:     while ∂cascade[σ] ≠ ∅ do
15:         τ ← Youngest(∂cascade[σ])
16:         if partner[τ] = ∅ then
17:             return
18:         else
19:             cascade[σ] ← cascade[σ] + cascade[partner[τ]]
20:         end if
21:     end while
22: end function

23: function Youngest(c)
24:     i ← ∞
25:     for Simplex σ in the simplicial chain c do
26:         if σ has a smaller index than i then
27:             i ← index[σ]
28:         end if
29:     end for
30:     return i
31: end function
```

Having rephrased the algorithm in terms of a matrix reduction scheme, we now only need to assure ourselves that the cascade operations do not change the boundary class. This is handled by the following lemma.

Lemma 4.19 (Correctness of cascade operations). By adjusting the cascade of a simplex σ in Line 19 of Algorithm 5, the homology class of σ does not change.

Proof. First of all, we note that τ, the 'youngest' simplex, must by necessity be a positive simplex. This is an inductive invariant that is maintained throughout the algorithm. We assume that τ has a partner. Else, the cascade will not be modified. Since τ is a positive simplex, its partner is negative. In particular, since $\tau \in \partial\text{cascade}[\sigma]$, the dimensions of its partner and σ are compatible. Furthermore, since the partner of τ destroys a homology class, we only add boundaries to the cascade of σ. Adding a boundary does not change the homology class by definition. ∎

The algorithm uses coefficients from $\mathbb{Z}_2$. It is also applicable for other coefficients from fields or, more generally, *principal ideal domains* [11, p. 396]. Zomorodian and Carlsson [410] state that the new algorithm may be cast into the same framework as the old algorithm [409], but requires proofs that are slightly more involved. We thus only use $\mathbb{Z}_2$ coefficients in this thesis. Their utility for data analysis has been confirmed by Zomorodian [406, pp. 56–57].

Choosing a filtration

A common filtration is the *distance filtration* that we already encountered in Definition 4.12. Given a distance measure dist on the data, we assign each 0-simplex a weight of 0, while each 1-simplex $\{i, j\}$ is assigned the weight $\operatorname{dist}(p_i, p_j)$, i.e. the distance between the i^{th} and the j^{th} data point. Higher-dimensional simplices are then assigned the maximum of the distance values stored in their faces. Calculating persistent homology with the distance filtration then corresponds to investigating the scale behaviour of data. The persistence in this case directly reflects the scales upon which certain appear or disappear.

If a scalar-valued function $f\colon \mathbb{D} \to \mathbb{R}$ is available, we can extend its values onto the simplicial complex by assigning each 0-simplex the corresponding function value and each higher-dimensional simplex either the maximum or the minimum value of its vertices. The ensuing filtrations are called the *sublevel set filtration* and the *superlevel set filtration*, respectively. Other filtrations are possible, as well. Carlsson [67] refers to using arbitrary scalar functions on a data set as *functional persistence*. He suggests studying their behaviour in order to learn more about the structure of the data. Another analogy is to refer to different filtrations as 'camera lenses' [253] through which analysts may focus their observations. Examples comprise functions based on the *singular value decomposition* (SVD) [140] of special quantities on the

data, density functions, and auxiliary topological descriptors such as graph Laplacians [102]. In Chapter 7, Section 7.5, p. 170 ff., we will define and evaluate several functions that are suitable for data analysis.

Performance improvements

A drawback for all calculations that we have seen so far is the large amount of memory that is required for storing the simplicial complex. The calculation itself has a worst-case computational complexity of $\mathcal{O}(n^3)$, where n is the number of simplices in the simplicial complex. In practice, however, the algorithm tends to have a sub-quadratic or even linear running time [409].

Still, persistent homology does not yet scale well to even moderately-sized data. Thousands of points comprise almost no problem to most clustering algorithms, but depending on their distance structure, persistent homology may well be infeasible on desktop machines. There is thus a large motivation for improving the performance of algorithms. The literature knows several different strategies. Cohen-Steiner et al. [106], for example, devised an algorithm for rapidly restoring filtration order, given small changes in the values of each simplex. Bauer et al. [30] presented modifications to the standard reduction algorithm that make it possible to compute persistent homology in parallel. In a follow-up publication [29], they presented further performance improvements that exploit the order in which simplices are paired.

Recently, some progress has been made by employing approximations to the Vietoris–Rips complex. Sheehy [334], for example, presented linear-sized approximations to the complex whose granularity can be controlled. Since persistent homology is an approximation to the homology groups of the input space, there is no need to keep the calculations as exact as possible. With some quality guarantees, the standard distance-based filtration can now be computed with significant speed-ups. In a similar vein, the usage of spectral sequences shows some promise for performance improvements [248].

4.5 Visualizing persistent homology

We have seen how to calculate the persistent homology of a data set, yielding a multi-scale description of its topology. This resulted in a set of tuples—or intervals—of the form $[c, d]$, with $c \in \mathbb{R}$ and $d \in \mathbb{R}_\infty$. We shall refer to these intervals as *persistence intervals* and present two common visualizations for them. In Chapter 5, we will derive an improved visualization that balances the relative merits of the other visualizations.

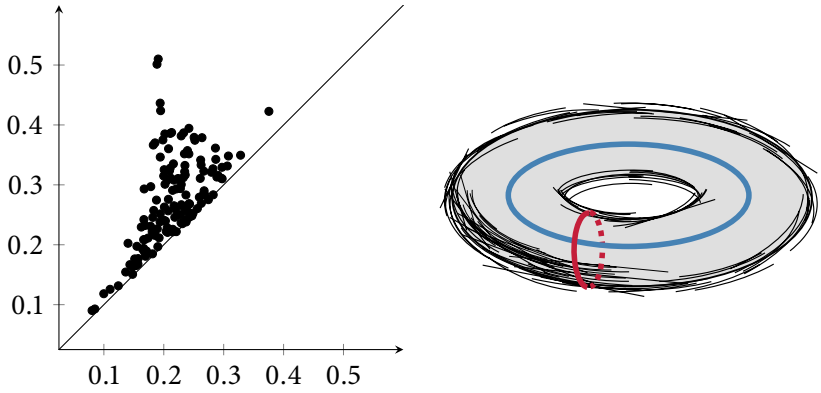


Figure 4.8: A persistence diagram of the 1-dimensional persistent homology of a synthetic torus. The cluster of points that is close to the diagonal is caused by the sampling process, while the two individual points correspond to the two loops of the torus, as shown on the right-hand side.

4.5.1 Persistence diagrams

A *persistence diagram* is in some sense the canonical visualization of persistence intervals. Given a set of intervals belonging to the same dimension, their persistence diagram is formed by drawing the point $(c, d) \in \mathbb{R}^2$ for each interval $[c, d]$. If $d = \infty$, we draw a point that is slightly above the boundary of the diagram. For filtrations with ascending values, the points in the persistence diagram fill up the region above the diagonal. Figure 4.8 shows a persistence diagram of the persistent homology in dimension 1 of a synthetic torus.

Definition 4.20 (Persistence of a point in a persistence diagram). Given a persistence diagram $\mathcal{D}$ and a point $(c, d) \in \mathcal{D}$, we define its *persistence* as

$$\operatorname{pers}(c, d) := |d - c|, \tag{4.21}$$

which is a value in $\mathbb{R}_\infty$. This is merely a reformulation of Definition 4.16 on p. 57, where we first defined persistence in a group-theoretic setting.



The persistence diagram was originally introduced by Edelsbrunner et al. [148] and has been employed extensively in subsequent publications [103, 104, 105, 106, 141, 142]. A family of persistence diagrams serves as a *small multiple* [367] fingerprint of the data set. In order to read this fingerprint, we observe that the distance of a point to the diagonal is correlated with its persistence, i.e. its scale. Points with a small distance to the diagonal are usually taken to be a sign of topological noise in a data set.

Lemma 4.21. The distance of a finite persistence pair $(a, b) \in \mathbb{R}^2$ to the diagonal is equal to its persistence, up to a factor.

Proof. Let d denote the distance to the diagonal. The pair (a, b) forms an isosceles triangle with the diagonal. Since the distance is measured using an orthogonal projection, the angle between the diagonal and the hypotenuse of the triangle is $\pi/4$. Hence, we have

$$\sin \frac{\pi}{4} = \frac{d}{|a - b|}, \tag{4.22}$$

which can be rearranged to

$$d = |a - b| \sin \frac{\pi}{4} = \frac{|a - b|}{\sqrt{2}}, \tag{4.23}$$

and the proposition follows. We thus conclude that it is justified to consider large distances from the diagonal to correspond to large persistence values. ∎

Properties of persistence diagrams A drawback of persistence diagrams is their tendency to appear cluttered and suffer from overplotting. Furthermore, following the 'gestalt' principles [378, 379], groups may be perceived incorrectly in persistence diagrams. In the case of the torus persistence diagram, for example, we perceive at least three points that appear to be outliers—only the single lone point may be considered an outlier, though. The pair of points actually corresponds to the two topological features of a torus in dimension 1. The eye of a viewer is thus drawn to the wrong cluster of points. In combination with having to estimate the distances of points to the diagonal—which works approximately at best—the persistence diagram is not the most intuitive visualization for showing persistence intervals. We shall use it nonetheless to explain distances and stability properties, mainly because of its excellent *data–ink ratio* [368, pp. 93–96].

Persistence diagrams & human perception Concerning the perception of differences, the persistence diagram uses human *pre-attentive processing* [365] to its advantage. Small-scale differences are not perceived as easily as large-scale changes, though. The persistence diagram groups points by 'scale similarity', meaning that topological features appearing and disappearing at similar scales are grouped. To enhance this display, additional stimuli such as colours could be introduced. They are known to override grouping by spatial proximity rather effectively [302] and thus permit the quantification of different attributes or the emphasis of existing ones, such as the persistence.

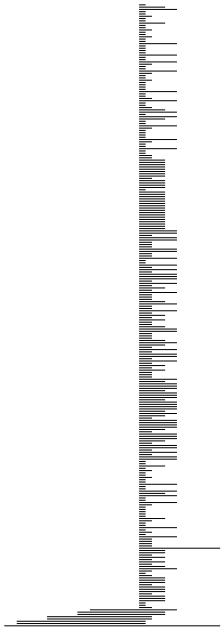


Figure 4.9: A persistence barcode of the 1-dimensional persistent homology of a synthetic torus. The order of intervals is unspecified, but it is a common practice to sort them by their creation value.

4.5.2 Persistence barcodes

To improve displaying scale information of each persistence interval, Ghrist [177] introduced the *persistence barcode* visualization. Here, each interval $[c, d]$ is drawn as an interval in the plane. Individual intervals are then stacked on top of each other. The result is a structure that visually resembles a barcode—hence the name. Persistence barcodes exploit the excellent human capability of judging lengths of lines [351, p. 15]. The eye is thus immediately drawn towards the largest scale information. This might be misleading, though, if topological features exist at different scales whose relative differences are very large.

Another disadvantage of the barcode is its scaling behaviour. Even for data sets containing only a few hundred points, the barcode gets very large. Figure 4.9 depicts an unembellished barcode of the 1-dimensional persistent homology of a synthetic torus data set. The two long bars represent the two generators in dimension 1. All other bars are caused by the sampling process, and while they still yield information about how the torus was constructed—in this case, the amount of 1-dimensional features implies that the torus has been sampled in circular slices—they are not as important as the two 'real' generators. In addition to the scaling problem, there is a degree of freedom in the sorting order of persistence intervals. While intervals could conceivably be sorted by their creation threshold, for example, it is not clear what the 'best' order is.

4.6 Quantifying topological similarity

So far, we have dealt with persistent homology on a *qualitative* level. We saw how to represent topological information by various visualizations that are useful for assessing differences and similarities between data sets. In the subsequent sections, we will pursue a more *quantitative* approach and describe several algorithms for computing the similarity between persistence diagrams. We will also discuss the stability of these calculations and, finally, compare topological and geometrical distances with respect to their robustness.

4.6.1 Distances between persistence diagrams

Persistent homology is equipped with two well-defined notions of distance. First, the *bottleneck distance* quantifies the maximum amount of disparity between two persistence diagrams. Second, the *Wasserstein distance* permits a more balanced calculation of dissimilarities. Prior to introducing these two distances, we first require several auxiliary definitions.

Definition 4.22 (L_∞-distance). Given two points $x, y \in \mathbb{R}^n$, their L_∞-distance is defined as the maximum difference over all dimensions, i.e.

$$\|x - y\|_\infty := \max\{|x_1 - y_1|, \dots, |x_n - y_n|\}, \tag{4.24}$$

which is a stable distance because it automatically suppresses the impact of small-scale perturbations.

We use the L_∞-distance instead of the usual Euclidean distance because it is better applicable for heterogeneous coordinates such as the ones that appear in persistence diagrams [8]. Furthermore, many *nearest neighbour* problems with arbitrary metrics may be reduced to nearest neighbour problems with the L_∞-distance by means of a properly-selected embedding [162].

Definition 4.23 (Bottleneck distance). Given two persistence diagrams $\mathcal{X}$ and $\mathcal{Y}$, their bottleneck distance is defined as

$$W_\infty(\mathcal{X}, \mathcal{Y}) := \inf_{\eta: \mathcal{X} \to \mathcal{Y}} \sup_{x \in \mathcal{X}} \|x - \eta(x)\|_\infty, \tag{4.25}$$

where $\eta: \mathcal{X} \to \mathcal{Y}$ denotes a bijection between the point sets of $\mathcal{X}$ and $\mathcal{Y}$ and $\|\cdot\|_\infty$ refers to the L_∞-distance between two points in $\mathbb{R}^2$. The bottleneck distance thus measures the maximum amount of displacement that is required to transform $\mathcal{X}$ into $\mathcal{Y}$. Since the cardinality of both diagrams is different in general, we require both diagrams to contain the orthogonal projections of their points to the diagonal. We then permit the bijection η to send a point in

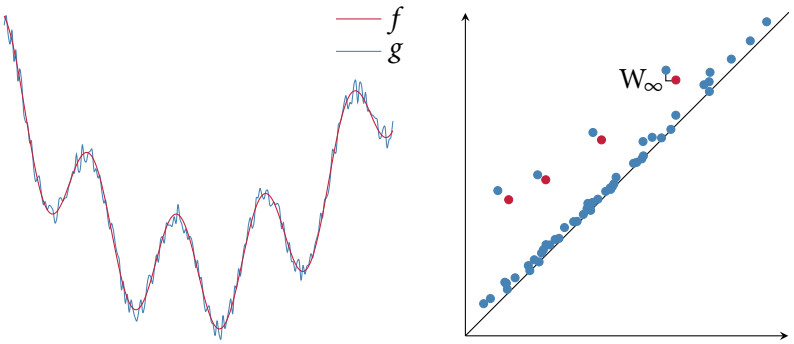


Figure 4.10: The bottleneck distance between two persistence diagrams. The two functions on the left-hand side are close in the Hausdorff sense. In their persistence diagrams, shown on the right-hand side, we see that the large-scale features (i.e. the minima–maxima pairs) of the original function are being retained by the perturbed function. The bottleneck distance is calculated using the largest difference found in the matching.

one diagram to its projection onto the diagonal. This indicates that the topological feature corresponding to the point remains unmatched.

Often, we represent a function or a topological space by means of more than one persistence diagram because we take higher-dimensional topological features into account. In this case, the bijection used in Equation 4.25 calculates the suprema in each dimension individually. Figure 4.10 illustrates the bottleneck distance for two simple functions. The strength of this distance is that it can be calculated even if the number of points in the two persistence diagrams varies considerably.

However, the bottleneck distance is not entirely robust with respect to outliers. A single large peak in the original function would increase the bottleneck distance between the persistence diagrams, as well as their Hausdorff distance. To decrease the influence of outliers, we may calculate the *Wasserstein distance* between persistence diagrams instead. It is slightly more expressive but requires a more complex calculation.

Definition 4.24 (Wasserstein distance). Given two persistence diagrams $\mathcal{X}$ and $\mathcal{Y}$, their p^{th} Wasserstein distance is defined as

$$\mathrm{W}_p(\mathcal{X},\mathcal{Y}) := \left(\inf_{\eta:\mathcal{X}\to\mathcal{Y}} \sum_{x\in\mathcal{X}} \|x-\eta(x)\|_\infty^p \right)^{\frac{1}{p}}, \tag{4.26}$$

where $\eta\colon \mathcal{X}\to\mathcal{Y}$ again denotes a bijection between the point sets of $\mathcal{X}$ and $\mathcal{Y}$. Just as for the bottleneck distance calculation, we permit η to send a point in one persistence diagram to its projection onto the diagonal in order to indicate topological features that remain unmatched.

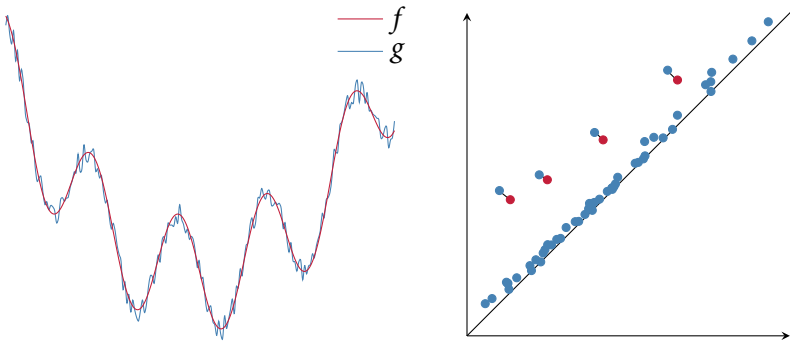


Figure 4.11: The Wasserstein distance between two persistence diagrams. The large-scale features are matched with each other (indicated by lines). The remaining features are matched to their projections onto the diagonal.

Formally, the Wasserstein distance converges to the bottleneck distance as the exponent p is increased, i.e.

$$\lim_{p \to \infty} \mathrm{W}_p = \mathrm{W}_\infty, \tag{4.27}$$

although in practice, the bottleneck distance is numerically more stable than calculating W_p for a very large p. The p^{th} Wasserstein distance is often preferable to the bottleneck distance because it is not insensitive to small-scale distances between $\mathcal{X}$ and $\mathcal{Y}$. By contrast, for W_∞, we only take the largest difference between the two diagrams into account.

In case we have associated multiple persistence diagrams to a space or a function, we sum over all individual infima in Equation 4.26 prior to calculating the p^{th} root. Figure 4.11 shows the matching of the previously-encountered persistence diagrams. We observe that only the large-scale topological features of both functions are matched with each other, while the remaining features of the second function remain unmatched. This indicates that the original function is more smooth than the perturbed function. In the calculation, p is a smoothing parameter. It is able to further reduce the effects of small-scale deviations. A value of $p = 2$ or $p = 1$ is sufficient for most applications. If not mentioned otherwise, we will be using the second Wasserstein distance W_2 throughout this thesis. The Wasserstein distance has numerous interesting properties, making it a suitable choice for many applications. In particular, it is known to include a large amount of geometrical information about the space it is calculated in. See Villani [376, pp. 110–111] or Villani [376, Chapter 6] for an overview.

Implementation & performance

Calculating W_∞ and W_p involves finding *maximum matchings* in weighted bipartite graphs. Algorithm 6 shows a pseudo-code description of the required steps. The matching can be calculated using the *Hungarian method* [226], which is also known as the Kuhn–Munkres

algorithm. It has a complexity of $\mathcal{O}\left(n^3\right)$, where n is the number of vertices in the graph. Recent research by Kerber et al. [220], based on previous work by Efrat et al. [153] showed that the complexity for calculating the bottleneck distance can be reduced to $\mathcal{O}\left(n^{1.5} \log n\right)$. The authors also report speed-up factors of 50–400 for the Wasserstein distance calculations, provided that 0.01-approximative solutions are sufficient. This makes both distance calculations very tractable, even for large persistence diagrams.

Algorithm 6: Calculating the Wasserstein distance between persistence diagrams

Require: Persistence diagrams $\mathcal{X}$ and $\mathcal{Y}$
function $\mathrm{W}_p(\mathcal{X},\mathcal{Y})$
  $(X, X_\perp) \leftarrow$ AddPoints$(\mathcal{X})$
  $(Y, Y_\perp) \leftarrow$ AddPoints$(\mathcal{Y})$
  Build a bipartite graph G with vertex set $X \cup Y_\perp \times Y \cup X_\perp$.
  for Edge $e = (u, v) \in G$ **do**
    $w_e = \begin{cases} \|u - v\|_\infty^p & \text{if } u \in X \text{ or } v \in Y \\ 0 & \text{else} \end{cases}$
  end for
  Calculate a maximum weighted bipartite matching on G.
  $c \leftarrow 0$
  for Edge e in the matching **do**
    $c \leftarrow c + w_e$
  end for
  return $c^{\frac{1}{p}}$
end function
function AddPoints$(\mathcal{D})$
  for Point $(x, y) \in \mathcal{D}$ **do**
    $X \leftarrow X \cup \{(x, y)\}$
    $X_\perp \leftarrow X_\perp \cup \left\{\frac{1}{2}(x + y, x + y)\right\}$
  end for
  return $(X, X_\perp)$
end function

Other distance measures

Recent work in persistent homology resulted in multiple new dissimilarity measures for persistence diagrams that are not necessarily metrics in a mathematical sense. Cerri et al. [83], for example, focus on improving the performance of bottleneck distance calculations by defining multi-scale approximations to this distance. Reininghaus et al. [306] developed a stable kernel that permits the comparison of persistence diagrams in a Hilbert space setting, making it possible to use persistence diagrams in the context of machine learning. Moreover, Reininghaus et al. [306] proved that the Wasserstein distance does not lead to a valid kernel in the sense of Hilbert spaces. Chen et al. [98] use discretized persistence diagrams and *kernel density*

estimation (KDE) [341] to perform clustering and two-sample tests. It is unclear whether this approach is stable, though. Bubenik [60] created a new summarizing description of persistence diagrams, the *persistence landscape*. He was able to show that persistence landscapes are defined in a Banach space, making the notion of random variables or a 'mean' diagram well-defined. Distances between persistence landscapes turn out to be stable summarizing statistics—they can be shown to yield lower bounds for the bottleneck distance and the Wasserstein distance. This implies that they are less expressive than these distances. On the other hand, they can be computed in linear time. Carlsson et al. [72] developed a pseudo-metric that is applicable for persistence barcodes. It was subsequently refined by Collins et al. [109] and has proven useful in an image analysis context. Adcock et al. [1], however, report that the pseudo-metric is very sensitive towards persistence barcodes with different cardinalities.

4.6.2 Stability

We have described different ways of calculating distances between persistence diagrams. From the persistence diagram construction, at least in the 0-dimensional case, it is clear that a small perturbation of the minima and maxima of the function will only result in a small perturbation of the corresponding persistence diagram. Stability—or robustness under noise—is a very important property to have. Our distance measures should work regardless of a small amount of noise, which real-world data inevitably suffer from. In the following, we will first take a look at the stability of the distance calculations with respect to the Hausdorff distance between two functions. Next, we will analyse stability in a more general setting by means of the Gromov–Hausdorff distance.

Bottleneck & Wasserstein stability

We will now take a look at how the intuition of stability in the 0-dimensional case may be formalized. It turns out that a perturbation of two functions f and g influences the distance between their persistence diagrams in a stable manner. We recall Definition 4.13 on p. 55 concerning monotonic functions on simplicial complexes. This permits us to state the first theorem concerning the stability of the bottleneck distance between persistence diagrams.

Theorem 4.25 (Bottleneck stability of persistent homology). Let K be a simplicial complex and $f\colon \mathrm{K} \to \mathbb{R}$, $g\colon \mathrm{K} \to \mathbb{R}$ two monotonic functions. For every dimension p, the bottleneck distance between the corresponding persistence diagrams is bounded from above by the maximum distance between the functions, i.e.

$$\mathrm{W}_\infty(f, g) \leq \|f - g\|_\infty, \tag{4.28}$$

where $\|f - g\|_\infty := \sup_x \|f(x) - g(x)\|$ is the supremum distance between the two functions according to Definition 4.22 on p. 68. Hence, any noise that does not influence the L_∞-distance will not influence the bottleneck distance calculation.

Proof. This was proven by Cohen-Steiner et al. [104], using a previous result by Robins [319]. The basic idea of the proof involves bounding the largest width of an empty box around points in both diagrams. ■

The preceding theorem also implies stability in the Hausdorff sense because the Hausdorff distance is always a lower bound for the bottleneck distance. While Theorem 4.25 only holds for functions on simplicial complexes, we can increase the scope of the theorem. By assuming that the functions f and g are *tame*, in the sense that they only have finitely many critical values, a similar theorem holds for arbitrary functions $f\colon \mathbb{X} \to \mathbb{R}$ and $g\colon \mathbb{X} \to \mathbb{R}$ on any triangulable topological space $\mathbb{X}$. Assuming the manifold hypothesis holds for an input data set, this theorem may also be considered the fundamental theorem of topological data analysis, as it permits us to calculate persistent homology and precisely know the stability properties of our approximations. It is somewhat surprising that we are able to obtain similar results for the Wasserstein family of distances. Since the p^{th} Wasserstein distance uses all points in a persistence diagram, the effects of noise are not necessarily mitigated. However, in case we restrict ourselves to Lipschitz-continuous functions, we can obtain useful bounds.

Definition 4.26 (Lipschitz continuity). Let $\mathbb{X}$ and $\mathbb{Y}$ be two topological spaces with metrics $\text{dist}_\mathbb{X}$ and $\text{dist}_\mathbb{Y}$. A function $f\colon \mathbb{X} \to \mathbb{Y}$ is *Lipschitz-continuous* if there is a constant $L_f \in \mathbb{R}$ with $L_f \geq 0$ such that

$$\text{dist}_\mathbb{Y}(f(x_1), f(x_2)) \leq L_f \, \text{dist}_\mathbb{X}(x_1, x_2) \tag{4.29}$$

for all $x_1, x_2 \in \mathbb{X}$. We assume that L_f has been chosen to be minimal and will refer to it as the *Lipschitz constant* of f. Moreover, we will refer to f as a *Lipschitz function*. The Lipschitz constant L_f controls how the norms are changed when going from $\mathbb{X}$ to $\mathbb{Y}$.

To state the theorem about Wasserstein stability, we require some technical definitions that are due to Cohen-Steiner et al. [105]. The basic idea is to define a measure that permits us to bound the amount of topological variation a function may exhibit.

Definition 4.27 (Degree-k total persistence). Let $k \in \mathbb{R}$ with $k \geq 0$. Furthermore, let f be any function and $\mathcal{D}$ its corresponding persistence diagram. The sum of all k^{th} powers of the persistence values of points in $\mathcal{D}$, i.e.

$$\text{Pers}_k(f) := \sum_{(c,d)\in\mathcal{D}} \text{pers}(c, d)^k, \tag{4.30}$$

is called the *degree-k total persistence of f*. In the previous equation, $\operatorname{pers}(c, d)$ refers to the persistence value described by Definition 4.20 on p. 65. We will re-encounter total persistence in Chapter 9, where we will use it to describe the geometrical–topological variation of a function. Here, we require total persistence to distinguish a certain class of metric spaces.

Definition 4.28 (Bounded degree-k total persistence). We say that a metric space $\mathbb{X}$ implies bounded degree-k total persistence if there is a constant $C_{\mathbb{X}}$ such that

$$\operatorname{Pers}_k(f) \leq C_{\mathbb{X}} \tag{4.31}$$

is satisfied for every tame function $f \colon \mathbb{X} \to \mathbb{R}$ with a Lipschitz constant $\mathrm{L}_f \geq 1$.

With these definitions, we may formulate a stability theorem for the Wasserstein distance W_p. The proof is due to Cohen-Steiner et al. [105], who also proved the stability of the degree-k total persistence as defined above.

Theorem 4.29 (Wasserstein stability of persistent homology). Let $\mathbb{X}$ be a triangulable, compact metric space that implies bounded degree-k total persistence for some $k \geq 1$. Furthermore, let $f \colon \mathbb{X} \to \mathbb{R}$ and $g \colon \mathbb{X} \to \mathbb{R}$ be two tame Lipschitz functions. The p^{th} Wasserstein distance between the persistence diagrams of f and g is then bounded, i.e.

$$\mathrm{W}_p(f, g) \leq C^{\frac{1}{p}} \|f - g\|_\infty^{1 - \frac{k}{p}}, \tag{4.32}$$

for all $p \geq k$ and $C \coloneqq C_{\mathbb{X}} \max\left\{\mathrm{L}_f^k, \mathrm{L}_g^k\right\}$. Hence, if we know the Lipschitz constants as well as the total persistence bound, we may precisely calculate the largest perturbation under which the Wasserstein distance will remain unchanged.

Requiring Lipschitz functions does seem rather restrictive at first. However, Lipschitz continuity turns out to be a natural condition that holds for many classes of functions. The Laplace–Beltrami operator [385, pp. 220–222], for example, which forms the basis for *heat kernel signatures*, can be shown to be a Lipschitz function [269]. Similarly, many centrality measures for graphs that we will encounter later on are Lipschitz functions [332].

Gromov–Hausdorff stability

Recently, Chazal et al. [88] were able to provide another insight into the stability of filtrations. They were able to show that the bottleneck distance between two filtrations is a lower bound for the *Gromov–Hausdorff distance* d_{GH} of two metric spaces. We first need to define a particular family of mappings between metric spaces.

Definition 4.30 (Isometry). Let $\mathbb{X}$ and $\mathbb{Y}$ be two metric spaces with corresponding metrics $d_{\mathbb{X}}$ and $d_{\mathbb{Y}}$. A map $f\colon \mathbb{X} \to \mathbb{Y}$ is an *isometry* if

$$d_{\mathbb{Y}}(f(a), f(b)) = d_{\mathbb{X}}(a, b) \tag{4.33}$$

holds for all $a, b \in \mathbb{X}$. Isometries are always injective—mapping two points a, b to the same point would immediately violate the metric property of $d_{\mathbb{Y}}$.

Isometries commonly occur when embedding a space in another space. Since they do not change the metric, they are sometimes also referred to as *congruence transformations*. The definition permits us to define the stability in the Gromov–Hausdorff sense.

Theorem 4.31 (Gromov–Hausdorff stability). For two metric spaces $\mathbb{X}$ and $\mathbb{Y}$ as defined above, the Gromov–Hausdorff distance between the two spaces is bounded from below by the bottleneck distance between the corresponding filtrations:

$$W_\infty(\mathbb{X}, \mathbb{Y}) \leq \mathrm{d_{GH}}\left((\mathbb{X}, d_{\mathbb{X}}), (\mathbb{Y}, d_{\mathbb{Y}})\right) \tag{4.34}$$

Formally, $\mathrm{d_{GH}}$ is defined as the smallest Hausdorff distance over all possible isometric embeddings of the two spaces $\mathbb{X}$ and $\mathbb{Y}$, i.e.

$$\mathrm{d_{GH}}(\mathbb{X}, \mathbb{Y}) := \inf\left\{ \mathrm{d_H}\left(f(\mathbb{X}), g(\mathbb{Y})\right) \mid f\colon \mathbb{X} \to \mathbb{Z}, g\colon \mathbb{Y} \to \mathbb{Z} \text{ isometries}\right\}, \tag{4.35}$$

where $\mathrm{d_H}$ denotes the Hausdorff distance and $\mathbb{Z}$ is an arbitrary metric space that does not refer to the field of integers.

The Gromov–Hausdorff distance is of particular interest in shape matching [56, 265] because it is invariant under isometries. This is desirable because it permits that a shape be rotated without changing its similarity to other shapes. As a consequence of the stability result from above, even small deformations of a shape will result in filtrations that are very close to each other. The disadvantage of the Gromov–Hausdorff distance is its computational intractability. In contrast to the Wasserstein and the bottleneck distance, where we only required bijections, we cannot easily enumerate all isometries between metric spaces. For practical calculations, the distance is thus often approximated [266].

4.6.3 Comparing topological & geometrical distances

In the context of *shape matching* and *shape retrieval*, topological distances are already known to be able to quantify different properties of a function than traditional *bag-of-features* models [246, 306]. In the following, we shall briefly outline some of their advantages over the more traditional distances in function spaces.

The L_∞-distance and the L_p-distances (with $p = 2$ more often than not) are commonly used in functional analysis to quantify the similarity of two functions f and g. Assuming that $f\colon \mathbb{D} \to \mathbb{R}$ and $g\colon \mathbb{D} \to \mathbb{R}$ are defined over the same domain $\mathbb{D} \subseteq \mathbb{R}$, we have

$$\operatorname{dist}(f,g)_{L_\infty} := \|f-g\|_\infty = \sup_{x\in\mathbb{D}} |f(x)-g(x)| \tag{4.36}$$

and

$$\operatorname{dist}(f,g)_{L_p} := \left(\int_{\mathbb{D}} |f(x)-g(x)|^p \,\mathrm{d}x\right)^{\frac{1}{p}}, \tag{4.37}$$

respectively. For higher-dimensional domains in some $\mathbb{R}^d$, the calculations work similarly. In the discrete case, f and g are often represented using a grid. The neighbourhood definitions of these grids become progressively costly to approximate and define with increasing dimensionality. Moreover, if f and g have different domains $\mathbb{D}_f$ and $\mathbb{D}_g$ with $\mathbb{D}_f \cap \mathbb{D}_g \neq \varnothing$, interpolations are required—and they may quickly become prohibitive for dimensions $d \gg 3$. Finally, if no grids are given, the evaluation of Equation 4.36 and Equation 4.37 requires spatial interpolation techniques such as *Kriging* [119, pp. 119–143] or *radial basis functions* [62].

One-dimensional test case

To compare the quantitative and qualitative behaviour of different distance functions, we prepared a simple one-dimensional test data set. The data set contains a one-dimensional test function whose y-values we perturbed by adding normal-distributed noise with $\mu = 0$ and $\sigma = 0.15$. Hence, the perturbed function has the form

$$f'(x) := f(x) + \mathcal{N}(0, 0.15), \tag{4.38}$$

which is a standard way of stating that the noise in a function follows a normal distribution. We do not perturb the x-values of the functions because it would put the function space distances at an obvious disadvantage. Since L_∞ and L_p are evaluated on the domain of the function, applying offsets in x-direction only results in increasing the distances between the functions. Topological distances, by contrast, are invariant to changes in their domain.

Results

A useful distance measure should either reflect the noise distribution in the data or ignore it altogether. Since the noise is following a normal distribution, the distance calculation—being in essence a calculation of the *differences* between perturbed functions—should also follow a normal distribution. In the following, we will hence assess the *normality* of these differences. We refrain from reporting the results of established normality tests, such as the

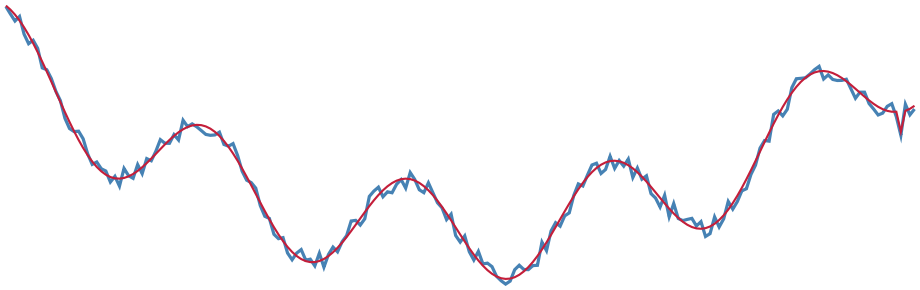


Figure 4.12: A one-dimensional test function and its perturbed variant. The original function ● and the perturbed function ● have the same large-scale features. A distance function should not add any additional noise to the calculations.

Shapiro–Wilk test because they are not easy to interpret. Instead, we show histograms and compare their shape—both visually and computationally—to that of a normal distribution. For the histogram calculation, we use the Freedman–Diaconis rule [168].

Figure 4.13 shows the histograms of the different distance measures. We first observe that the L_∞-distance is highly-dependent on the largest perturbation. It considers all functions to be very dissimilar from each other, leading to a highly-skewed distribution. For the bottleneck distance W_∞, it is just the other way round. By focusing on pairing topological features—maxima and minima—in the functions, the distance has a peak towards zero, indicating that most of the functions are being considered very similar. Finally, the L_2-distance and the Wasserstein distance W_2 exhibit a concentration of values around 0.5. The *kurtosis* of the W_2 histogram is 3.09, which is very close to the kurtosis value of 3.0 of a normal distribution. By contrast, the kurtosis of the L_2 histogram is 3.40. We hence obtain the best approximation of the noise profile by the Wasserstein distance W_2. This does not imply that function space distances are generally unsuitable. It merely illustrates that distances based on the topological approximation of data have their own merits and may be advantageous in many cases, in particular in the context of data analysis.

4.7 Discussion

This chapter introduced *persistent homology*, the main concept used in this thesis. We encountered different computational strategies for specific cases such as one-dimensional data, as well as for generic cases. The basis for all these computations is an approximation of the connectivity of a data set, calculated using geometrical complexes such as the Vietoris–Rips complex $\mathcal{V}_\epsilon$. We described an algorithm for obtaining a Vietoris–Rips complex from multivariate data, provided a distance measure is available. Following this, we discussed two strategies for calculating persistent homology. We also encountered two standard ways—*persistence*

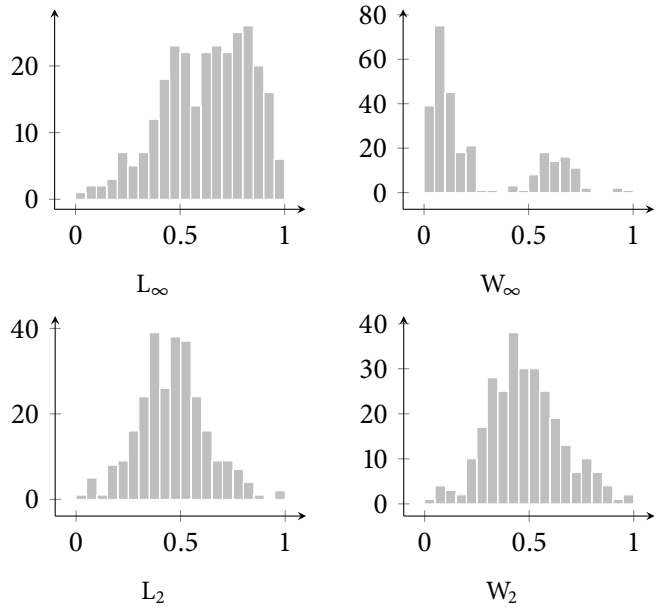


Figure 4.13: Histograms showing the distances between the test functions. All distances have been scaled to $[0,1]$ in order to make the comparisons fair. We can see that the second Wasserstein distance W_2 is the most sensitive of all distance measures because it best approximates a normal distribution. The L_2-distance yields similar results but its shape differs slightly from the shape of a normal distribution.

diagrams and *persistence barcodes*—of visualizing the results of the persistent homology calculation. While both visualizations are used extensively in the literature, we also saw that they suffer from drawbacks, most notably overplotting and scale issues. Moreover, this chapter introduced several ways of quantifying the topological similarity of functions. We encountered two notions of distance, the *bottleneck distance* and the *Wasserstein distance*. Both distances assess similarity by means of persistence diagrams and require solving an assignment problem. We gave a simple pseudo-code implementation to calculate distances. Furthermore, we compared topological distances with function space distances. We found out that even in the one-dimensional case, topological distances—in particular the Wasserstein distance W_p—have many beneficial properties beyond the common invariance properties of algebraic topology. In addition, we demonstrated that topological distances perform well under noisy conditions and are able to extract a correct noise profile from perturbed data.

✦

Nonetheless, there is a large amount of future work yet to be done. For example, instead of multi-scale approaches [83] for approximating topological distances, we suggest investigating different optimization strategies for the persistence diagram distance calculations.

A *nearest neighbour heuristic* for obtaining matches during the calculations of W_∞ or W_p may prove to be useful. It would be interesting to study the properties of such a heuristic. Unfortunately, in contrast to the *maximum weight matching* in bipartite graphs, for example, there are no approximation guarantees for greedy heuristics. Preliminary experiments by the author seem to imply that a simple heuristic based on nearest neighbour estimations only overestimates the correct distance by about 10% on average, while having much lower computational requirements.

Finally, so far we shied away from evaluating which functions are suitable as descriptors on a space. In the experiments presented in this chapter, we were able to make direct use of the sublevel or superlevel sets of our function because we were analysing the topology of the function itself. A hitherto-unanswered question concerns the use of useful shape descriptors for analysing multivariate data. We will come back to this aspect in the subsequent chapters.

Part I

Visualizing qualitative topological information

The first part of this thesis presents novel visualization techniques for *qualitative topological information* of data. Serving as an expressive 'perceptual fingerprint', we shall see how these techniques permit discovering and comparing intrinsic properties of multivariate data. The methods described in the subsequent chapters thus support and augment the *exploratory data analysis* workflow.

5 Topological fingerprints in cluster analysis

Clustering remains one of the most relevant techniques for analysing multivariate data sets. It is common to apply a clustering algorithm to one's data, extract the clusters, and analyse them separately. The shape information of clusters has largely been ignored in the literature so far due to the complexity of high-dimensional structural information. However, in almost any clustering analysis scenario, users want to know how the individual clusters differ in their structures. Persistent homology is ideally suited to serve as a shape descriptor. Its robustness with respect to small perturbations and changes lets it focus on large-scale differences.



In this chapter, we will develop a novel workflow for multivariate data analysis that employs persistent homology at its core. Qualitative topological information is visualized using *persistence rings*, a novel visualization technique of persistent homology that combines the relative advantages of persistence diagrams and persistence barcodes. In addition to a discussion about topological features in a synthetic data set, the efficacy of this approach is demonstrated by the analysis of highly-complex data sets from cultural heritage that are not amenable to standard analysis methods. This chapter is based on a previous publication [318] by the author.

5.1 Persistence rings

Recall that the result of persistent homology calculations was a set of intervals of the form $[c, d]$, with $c \in \mathbb{R}$ and $d \in \mathbb{R}_\infty$. In Chapter 4, we already encountered two standard methods for visualizing these persistence intervals, namely *persistence diagrams* and *persistence barcodes*. Drawing from the strengths of both existing persistence visualizations, the author [318] introduced *persistence rings*, a radial visualization of persistence intervals. It uses scalable circular segments that are arranged radially in order to remain compact while still retaining the scale information. Just like a persistence diagram, it serves as a *small multiple* [367] fingerprint of a data set. It does not suffer from any occlusion problems, though, because it depicts all topological features at unique positions.

Basic idea

To combine the advantages of persistence barcodes and persistence diagrams, namely the good interpretability and the compact layout, a circular arrangement of persistence intervals is advantageous. Each persistence interval $[c, d]$ is thus assigned an annular sector from radius c to radius d. If $d = \infty$, we use a radius that is larger than the remaining radii. Hence, the extent of each slice encodes the persistence $\text{pers} = d - c$, making it easy to judge accurately according to psychophysics [351, p. 15]. The persistence of each slice is furthermore encoded by the colour of each segment, using a continuous colour map  , which employs darker colours to indicate larger persistence values. The circular arrangement results in a compact display for even larger amounts of persistence intervals. Different radial arrangements can be used to ensure that the visualization is stable—in the sense that small changes in the persistence intervals only result in small changes to the arrangement. Moreover, modifying the placement of intervals enables us to exploit *pre-attentive processing* [365] again. The persistence rings are thus capable of guiding the focus of users directly to parts of the data where the topology is significantly different.

Algorithmic issues

Since we are working with a circular layout, we have two degrees of freedom for placing an annular sector. First, we have an opening angle θ that determines the size of the sector. Second, we have an angular offset ϕ that determines the radial position of the sector. There are different ways of selecting these angles, depending on the desired view on the persistence intervals. Following æsthetics criteria from graph drawing [300], we need to ensure that different sectors do not overlap. The angle an annular sector occupies should indicate its relevance to some extent. This global optimization problem is made more complicated by the fact that intersections between intervals are intransitive—meaning that if intervals x and y intersect and y intersects with another interval z, it does not necessarily follow that x and z have a common intersection as well.

A heuristic

Next, we will describe a useful heuristic for the layout. The heuristic guarantees that the sectors will be placed without overlaps. It then proceeds with two different—and somewhat incompatible—objectives. First, the heuristic tries to use all available space as much as possible. Second, it shall ensure that the size of each segment corresponds to its persistence value. The algorithm then assigns an angle $\theta \in [0, 2\pi)$ and an offset ϕ for each annular sector. We experimented with different algorithms until we found one that is a good compromise

between running time, appearance, and distortion. The idea of the heuristic is to place persistence intervals in order of increasing persistence. This is motivated by the following insight: Since an interval with an extremely high persistence value will block a large part of the available radius of a data set, its opening angle does not have to be very large. An interval with a low persistence value, on the other hand, needs to have a larger opening angle to be noticeable. For each interval, the heuristic determines the angular portion of the circle that is still available for placement. To this end, it employs an *interval tree* [37, Chapter 10.1, pp. 220–226] data structure. This tree permits efficient queries on a set of intervals to determine their intersections. It is highly-efficient, requiring only $\mathcal{O}(\log n + m)$ time per query, where n is the amount of intervals in the tree and m is the number of intervals returned by the query. Algorithm 7 contains a pseudo-code description of the heuristic, while Figure 5.1 shows the persistence ring of the 1-dimensional persistent homology of a torus.

Algorithm 7: Persistence ring calculation

Require: Set $\mathcal{I}$ of persistence intervals
1: **function** PLACEINTERVALS($\mathcal{I}$)
2: Calculate an interval tree $\mathcal{T}$ from $\mathcal{I}$.
3: Partition $\mathcal{I}$ into *finite* and *semi-finite* intervals.
4: Sort the finite intervals by increasing persistence.
5: Sort the semi-finite intervals by increasing creation time.
6: **for** Interval I $\in \mathcal{I}$ **do**
7: $\mathcal{N} \leftarrow$ FINDOVERLAPPINGINTERVALS($\mathcal{T}$,I)
8: n $\leftarrow$ 1, s $\leftarrow$ 0, o $\leftarrow$ 0
9: **for** Interval N $\in \mathcal{N}$ **do**
10: **if** ALREADYPLACED(N) **then**
11: o $\leftarrow \max(\text{o}, \text{N}.\phi + \text{N}.\theta)$
12: **else**
13: n $\leftarrow$ n + 1
14: s $\leftarrow$ s + N.persistence
15: **end if**
16: **end for**
17: $\alpha \leftarrow 2\pi -$ o, I.$\phi \leftarrow$ o
18: **if** I is finite **then**
19: I.$\theta \leftarrow \alpha \cdot$ I.persistence/s
20: **else**
21: $\theta(I) \leftarrow \alpha$/numNeighbours
22: **end if**
23: **end for**
24: **end function**
25: **function** FINDOVERLAPPINGINTERVALS($\mathcal{T}$, I)
26: Query the interval tree $\mathcal{T}$ to find all intervals that overlap with I.
27: **end function**

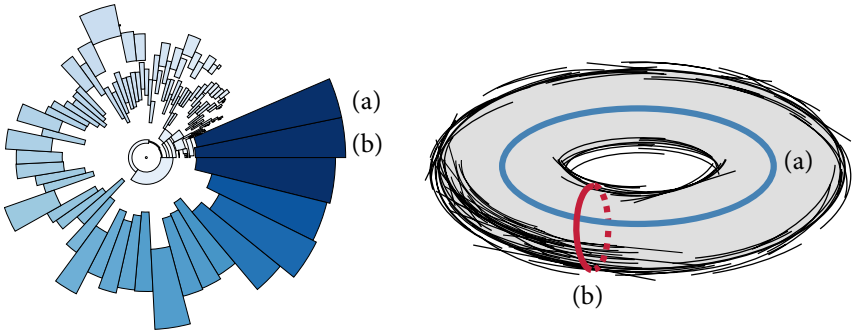


Figure 5.1: Persistence ring visualization of the 1-dimensional persistent homology of a synthetic torus. All sectors are coloured according to their persistence value. The two largest slices indicate the two features of infinite persistence. These features correspond to the two generators of the first homology group of the torus. The remaining features are created by the small circles that can be drawn on the torus longitudinally.

Comparing persistence rings

To show the benefits of the persistence ring visualization, we compare their 'visual performance' for random samples. We use a *ring torus* with parameters $R = 0.25$ and $r = 0.1$ as the underlying topological object, from which we sample $n = 500$ points. To ensure that the statistical properties of the torus are represented correctly, we use a rejection sampling approach [128], which is briefly described by Algorithm 8. We calculate the torus coordinates via

$$\begin{aligned} x &= \big(R + r\cos(\theta)\big)\cos(\psi) \\ y &= \big(R + r\cos(\theta)\big)\sin(\psi)\,, \\ z &= r\sin(\theta) \end{aligned} \tag{5.1}$$

where θ and ψ are the sets of angles obtained from the rejection sampling procedure. Figure 5.2 depicts some typical results of the sampling process, while Figure 5.3 on p. 88 shows different persistence visualizations for some of the samples. We only show 1-dimensional persistent homology classes. The persistence rings make it easy to show the structure of the data—the two different generators in dimension 1, for example, are readily visible in all instances of the data. In the sample shown in Figure 5.3c, points are spaced more regularly than in the other samples. As a consequence, the circular structure of the torus is visible at smaller scales already, leading to a decrease in the amount of topological features. This difference is only visible in the persistence rings—while at the same time, the persistence rings are the only visualization technique that highlight the similarity of the individual samples. By contrast, the persistence diagrams of the samples shown in Figure 5.3c and Figure 5.3d appear to suffer from an outlying point of low persistence. Our eye is immediately drawn to this point although it

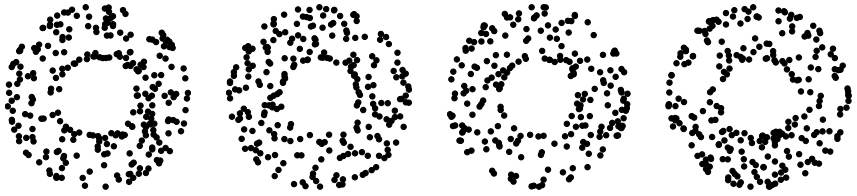


Figure 5.2: Projections of random samples from a torus. Each figure shows the projections of the sampled points to the xy-plane.

has no real significance for the general structure of the data. The effects of such points are mitigated by the persistence rings visualization.

Finally, we note that it is difficult to see the similarities between the different samples if we only make use of the persistence barcodes. Since barcodes are usually sorted by the creation time of the corresponding persistence interval, small changes in a pairing may result in large visual differences. This could conceivably be solved by investigating different orderings, similar to the issues with PCPs [9].

Algorithm 8: Torus rejection sampling

```
function TorusSamples(R, r, n)
    θ ← ∅
    ψ ← ∅
    for i in {1, ..., n} do
        x ← U(0, 2π)                          ▷ Draw from a uniform distribution
        y ← U(0, 1/π)                         ▷ Draw from a uniform distribution
        f ← (1 + r/R cos(x))/(2π)
        if y < f then
            θ ← θ ∪ {x}                       ▷ Add angle if condition is satisfied
            ψ ← ψ ∪ {U(0, 2π)}                ▷ Add angle from uniform distribution
        end if
    end for
end function
return (θ, ψ)
```

5.2 Combining topological analysis & clustering

We have seen that the persistence rings are useful descriptors of the topological activity found within data sets. In practice, our data sets often contain groups with different behaviours, making them amenable to *cluster analysis*. The field of *data mining*, of which cluster analysis is an integral part, is too large to be described in detail here. We refer the reader to standard textbooks [354, 402] or recent surveys [40, 211, 397] for more information.

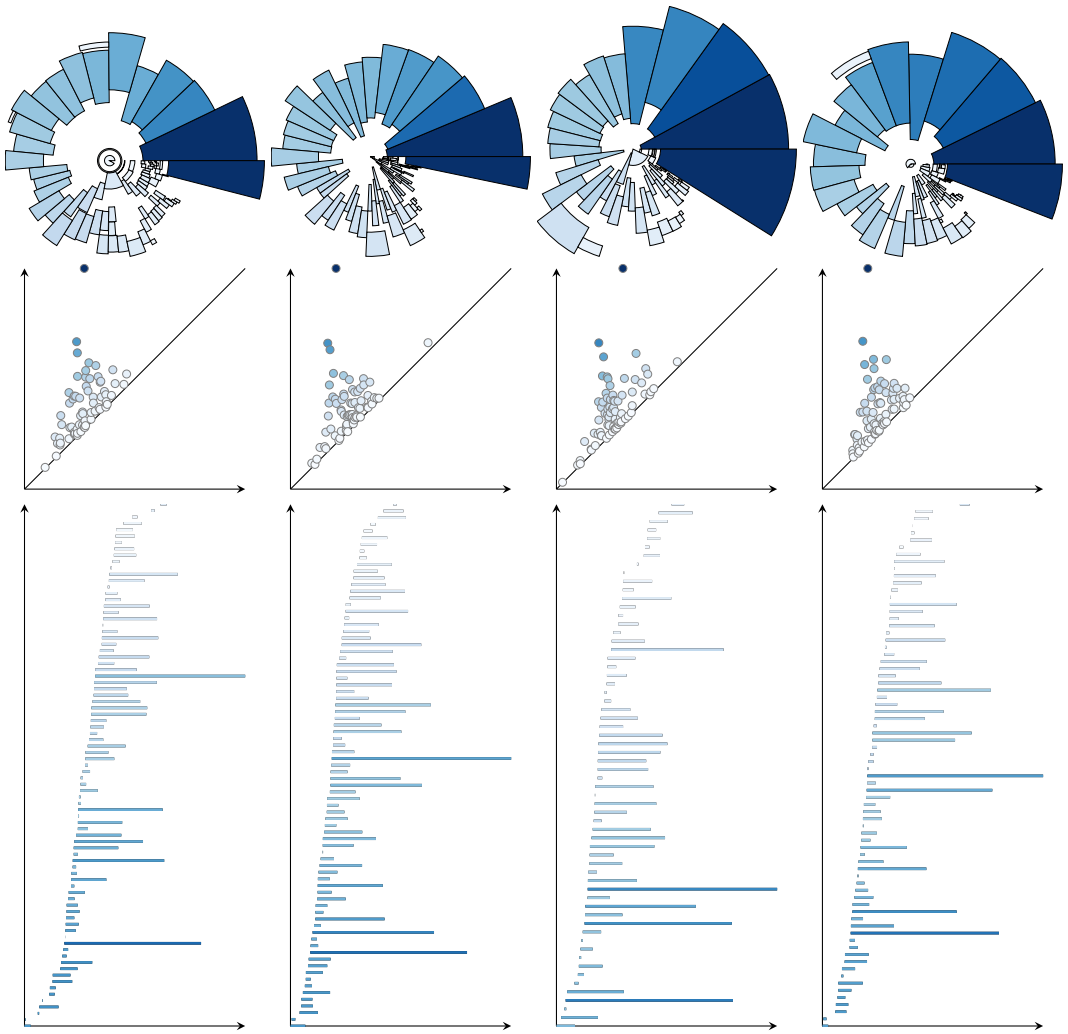


Figure 5.3: Comparing persistence rings of torus samples. Every column represents a single sample. The persistence rings make it easy to see that the samples have similar topological features.

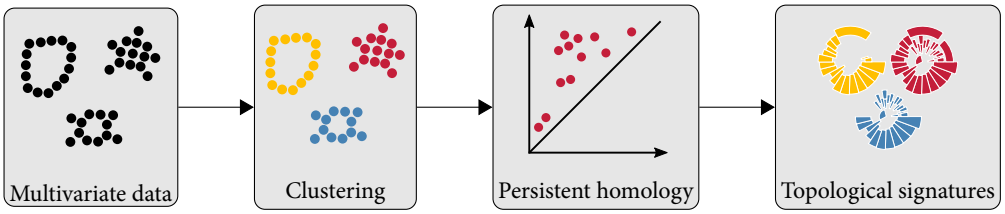


Figure 5.4: Multivariate data analysis workflow using topological signatures. Through interaction with the resulting topological signatures, users may study the shapes of their clusters.

Cluster analysis is often performed with the goal of predicting labels in a data set. Especially for experimental data, we either do not have this information available or may only obtain it in a tedious manner. Topological data analysis can help by providing a signature for each cluster. Instead of having to deal directly with the high-dimensional clusters—which may often not be possible due to dimensionality constraints—we may thus deal with the signatures. The *topological signature* of different clusters may be used to compare clusters in a qualitative manner, while keeping their multi-scale structure intact. We shall demonstrate later on how this information can be exploited. Figure 5.4 shows a graphical depiction of the proposed workflow. Next, we shall take a look at the required steps individually.

Obtaining clusters

Using clustering algorithms such as k-means [211], we first need to define clusters in our data. In practice, we often do not know what a suitable clustering algorithm for our data set will be. Later on—in Chapter 9—we shall see how to use persistent homology to simplify the selection of good clustering algorithms. For now, we assume that the user has selected a clustering of the data, for example by using auxiliary visualizations such as dendrograms [345].



For the analysis of cultural heritage data, as described in Section 5.6, we will make use of a density-based clustering algorithm because we do not have any reasonable assumptions about the shape of our clusters. The algorithm is based on concepts of persistent homology and thus integrates easily into our workflow—this does not prevent the use of standard clustering algorithms, though. The resulting clustering serves as a scaffold for the subsequent analysis. Instead of attempting to analyse labels in each cluster, we perform a qualitative analysis of the topological structure of individual clusters.

Calculating cluster signatures

We treat each cluster as a single input data set with no overlaps to the complete data set. Using a Rips graph with automated parameter selection, we may then calculate persistent homology on each of the clusters. This yields a set of persistence intervals, which we represent using the persistence rings we described earlier.

Comparing cluster signatures

Using the persistence rings as a topological signature, we may now compare the clusters among each other. This can be done in a qualitative manner, e.g. by looking at the amount of topological features of a certain persistence, or in a quantitative manner, e.g. by calculating distributions or distances in the space of persistence diagrams. We have implemented several interaction techniques, such as brushing+linking [63, 130], that permit users to link topological features to features in the data. The persistence rings also permit region and windowing queries, meaning that subsets of the persistence intervals may be selected for further analysis.

5.3 Persistence-based clustering

Prior to calculating persistent homology and topological signatures, our workflow from Figure 5.4 requires the definition of clusters in the data. This is a central step in the analysis of multivariate data. In the following, we present a clustering algorithm that combines density estimation and persistent homology. If no particular assumptions can be made about an input data sets, clustering approaches based on density estimation have proven to be very effective. One of the most successful clustering algorithms, DBSCAN [160], for example, is capable of finding cluster boundaries by changes in the density functions, which makes it capable of handling even complex cluster shapes. Recent works in visualization, e.g. by Oesterling et al. [286], also prove that density functions are meaningful descriptors for high-dimensional data sets.

The central idea behind any density-based clustering algorithm is that the data set has an underlying density distribution f that is unknown. Assuming that we have a way of estimating f, the clusters and the peaks—the maxima—of f coincide. This idea is a common strategy in many disciplines; Cheng [99], for example, refers to it as *mode-seeking*. In practice, mode-seeking is challenging because the density estimates exhibit a multitude of peaks, making it hard to define the thresholds for extracting clusters. We can circumvent this issue using persistent homology. By observing the changes the *superlevel sets* (see Definition 2.2 on p. 19) of the density function, we can assign each peak of the function a persistence value and only extract those peaks—i.e. clusters—that are sufficiently persistent. In the following, we

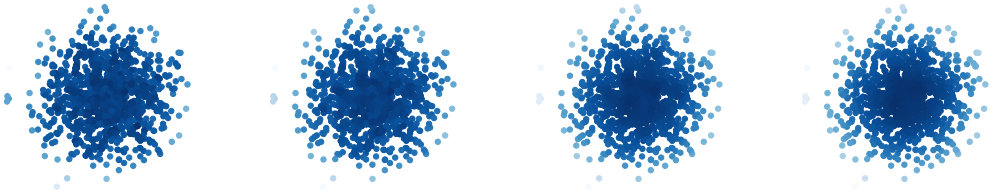


Figure 5.5: Example density estimates. The *distance to a measure* density estimator is extremely stable. There are almost no perceivable differences in the different point clouds. Later on, we will perform a more detailed analysis of the stability of the estimates.

present an improved and simplified version of an algorithm by Chazal et al. [91], which we use for the subsequent cluster analysis. The algorithm can be easily implemented and scales very well because it only requires a single pass through the data.

5.3.1 Density estimation

We first need a suitable density estimator for our data. Density estimation has a long tradition in statistics and it would go beyond the scope of this thesis to describe the field in detail. The reader is referred to the classic textbook by Silverman [341] for more information.

Distance to a measure

Following previous work by Chazal et al. [89] and Biau et al. [46], we use the *distance to a measure* density estimator. It requires a distance function $\mathrm{dist}(\cdot,\cdot)$, such as the Euclidean distance, and a neighbourhood parameter k. The estimator then calculates the mean squared distance to the k nearest neighbours of a point, i.e. we have

$$f(x) = -\frac{1}{k}\sqrt{\sum_{i=1}^{k} \mathrm{dist}^2(x, n_i)}, \tag{5.2}$$

where x refers to the query point and n_i to its i^{th} neighbour with respect to the distance measure. Given suitable data structures, such as the ones provided by FLANN [275], this estimator can be implemented efficiently. Its robustness under noise [89] makes it a reasonable choice for data analysis tasks. In the previous equation, the minus sign has been added to ensure that high values—i.e. values close to zero—correspond to dense areas in the data.

Figure 5.5 depicts the changes of the density estimates for varying k. A result of Biau et al. [46] states that the level sets extracted from the density estimator are extremely stable. Later on, in Section 5.5, we will analyse the stability of the estimator by means of a complex synthetic test data set.

Gaussian kernel

For smaller data sets, we can pursue a more classical approach and use a *Gaussian kernel estimator*. This estimator uses all available data points but scales their influence depending on the distance to the current query point x. The density around x is then estimated as

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} \exp\left(-\frac{d^2(x, x_i)}{\sigma} \right), \tag{5.3}$$

where σ can be used to control the smoothing of the estimates. The disadvantage of this estimator is that it does not scale well for larger data sets. If not specified otherwise, we use the *distance to a measure* estimator.

5.3.2 Peak estimation using persistent homology

Given a density estimator, such as the *distance to a measure* estimator described in the previous section, we now require an approximation of the connectivity of our data. The Rips graph $\mathcal{R}_\epsilon$, in conjunction with a suitable scale estimation algorithm, turns out to be a good choice here. Given $\mathcal{R}_\epsilon$ for some threshold ϵ, we extend the values of the density estimator to the edges of the graph by setting $f(u, v) = \max(f(u), f(v))$, meaning that the function value of an edge is the maximum of the density values of each of its vertices.

We could now calculate 0-dimensional persistent homology as described by Algorithm 3 on p. 54. The clusters that are being created using this approach will simply correspond to the connected components of $\mathcal{R}_\epsilon$—which is almost always not the desired result. Hence, we describe an improved and more efficient algorithm that incorporates user input. The algorithm has two phases, a *mode-seeking* phase and a *merging* phase.

Mode-seeking phase

In the mode-seeking phase, all potential modes (i.e. peaks) of $\mathcal{R}_\epsilon$ are being enumerated. We connect every vertex to its neighbour with the highest function value. If no such neighbour exists, the vertex is a potential mode. This phase results in a set of direct graphs, in which each edge shows the direction in which the function value increases. Hence, the edges created in this phase are akin to the *pseudo-gradient* calculation in discrete vector fields.

Merging phase

In the merging phase, we use a union–find data structure [113, Chapter 21] to keep track of the cluster associations of vertices. Each mode is assigned its own cluster at the beginning. We then traverse the vertices in descending order of their function values. At each vertex,

we traverse its neighbours and examine the values of their corresponding modes. So far, this procedure is fully equivalent to the 0-dimensional persistent homology calculation, as described by Algorithm 3 on p. 54. The difference lies in the way we perform the merges. Instead of always merging connected components when they are connected by an edge, we only perform a merge if the persistence of the corresponding modes is significant enough. To this end, we introduce a threshold parameter τ into the calculation. If the difference between the mode function values is less than or equal to τ, the mode is not deemed significant and merged with its neighbours. τ thus describes the minimum persistence—with respect to neighbouring modes—a peak must have until it is considered significant. Initially, $\tau = \infty$, which results in the ordinary 0-dimensional persistent homology calculation. Algorithm 9 on p. 93 describes a pseudo-code implementation of the clustering procedure.

Algorithm 9: Persistence-based clustering

```
    Mode-seeking phase
 1: Sort vertices of R_ε by decreasing function value.
 2: Initialize empty set of edges.
 3: for all Vertices v do
 4:     Connect v to its neighbour with the largest function value.
 5:     if v has no neighbour with a larger function value then
 6:         Mark v as a potential mode.
 7:     else
 8:         Store the edge in the set of edges.
 9:     end if
10: end for

    Merging phase
11: Initialize empty union–find data structure U.
12: for all Vertices v do
13:     if v is a mode then
14:         Add a new root entry to U.
15:     else
16:         f_v ← f(U.find(v))
17:         for all Neighbouring vertices w of v do
18:             f_w ← f(U.find(w))
19:             if |f_v − f_w| ≤ τ then
20:                 Use U to merge the component of w into the component of v.
21:             else
22:                 Use U to merge the component of v into the component of w.
23:             end if
24:         end for
25:     end if
26: end for
```

Parameter selection & stability

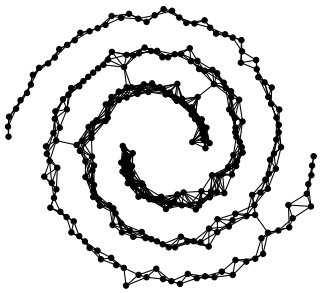
Having explained the algorithm, we now need to think about how to select suitable parameters. In Section 5.4, we will discuss generic heuristics for choosing the approximation parameter ϵ for calculating the Rips graph $\mathcal{R}_\epsilon$. Following this, we will discuss the stability of the density estimator by means of complex synthetic data in Section 5.5. The last parameter of the algorithm—the merge threshold τ—may be selected by looking at the persistence diagram of the clustering procedure. We can create this persistence diagram during the merging phase of Algorithm 9 by keeping track of the individual merges between vertices. Choosing τ has the effect of considering a topological feature to be noise if its persistence is less than or equal to τ. We may visualize this by a region in the persistence diagram—if we translate the diagonal by τ, every mode in the region enclosed by the diagonal and the translated line will be merged into the nearest peak. We shall discuss this by analysing example data in the next section.

Related work

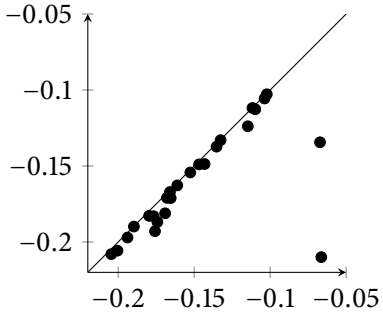
The 'tracking' of connected components, as described by the previous algorithm, is a very common idea that is often found in other contexts, such as *contour trees* [346] or *Reeb graphs* [387]. The analysis of the connectivity changes bears a close resemblance to the *join tree* [79].

5.3.3 An example

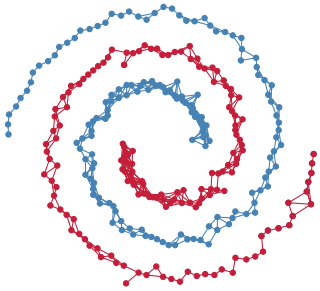
We use a data set containing two interlocked spirals to illustrate how the clustering algorithm works. This data set is a good showcase for density-based methods, as regular clustering methods, such as k-means, are incapable of separating the spirals. Figure 5.6 illustrates the clustering process. We first use the parameter selection algorithm from Section 5.4 to obtain a Rips graph $\mathcal{R}_\epsilon$, which is depicted in Figure 5.6a. We can see that there are some edges connecting both spirals. The persistence diagram that we obtain by running the clustering algorithm with $\tau = \infty$, is depicted in Figure 5.6b. Since the density estimator uses negative values, all points are located below the diagonal—this is somewhat different from the other persistence diagrams we have encountered so far. The diagram contains two points of high persistence, so we choose τ such that only those points will be considered 'real' peaks. After re-running the algorithm with $\tau \approx 0.05$, Figure 5.6c shows the decomposition of $\mathcal{R}_\epsilon$ into two clusters. The modified persistence diagram in Figure 5.6d contains the region corresponding to τ. We can see that all other modes are considered noise. Moreover, the colours of the two peaks indicate their corresponding connected component in the Rips graph.



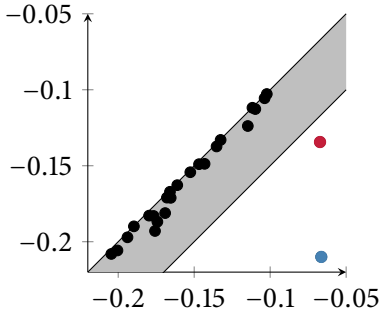
(a) Rips graph



(b) Persistence diagram with $\tau = \infty$



(c) Clustered Rips graph



(d) Persistence diagram with $\tau \approx 0.05$

Figure 5.6: An illustration of the persistence-based clustering algorithm. Starting from the Rips graph on the unclustered data (a), we obtain a persistence diagram (b) that shows two prominent peaks. All points are located below the diagonal because of the definition of the density estimator. After modifying the merge threshold τ so that only these two peaks are considered significant, we obtain two clusters (c). The modified persistence diagram (d) depicts the selected threshold.

Algorithm 10: Scale estimation based on Euclidean Minimum Spanning Trees

Require: Data set $\mathbb{X} = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$

```
function EstimateScaleMST(𝕏)
    Calculate a minimum spanning tree T using the pairwise point distances.
    Calculate the maximum edge length in T.
    return Maximum edge length
end function
```

5.4 Rips graph parameter selection

We have previously seen that the calculation of topological signatures requires a careful selection process for the scale parameter ϵ that is used in calculating persistent homology. Even though this part is crucial for any topological approximation, it has hitherto been largely ignored in the literature. Ideally, we want to have the smallest graph that still permits us to derive the relevant topological features. Previous approaches include a heuristic by Chazal et al. [91], who suggest using dendrograms from single-linkage clustering, in order to find suitable values for ϵ. No viable algorithm is provided, though, and the idea involves some guesswork. Users should examine the dendrogram and encounter scale information such that ϵ reaches a level in which the 'most relevant' structures are already clustered. In the context of scalar field topology, Correa and Lindstrom [115] allude to finding thresholds at extremely large scales by means of geometric graphs and pruning away undesired edges afterwards. They conclude that scales cannot be estimated in a reliable manner. In a subsequent publication [114], Correa and Lindstrom use *empty region graphs* [66] to obtain scale estimates for semi-automated spectral clustering. Their method still requires parameter selection for both the graphs and the local smoothing of scales. In light of these issues, it makes sense to propose a scheme that is geared towards scientific data sets. Here, we often experience a dense core structure of a data set [214], along with some outlying points. Given the assumption that a single manifold underlies a data set, it thus makes sense to choose ϵ large enough to obtain a Rips graph with a single connected component—or, failing that, at least a Rips graph whose connected components are as large as possible. We envision two different heuristics for obtaining suitable thresholds.

Minimum spanning tree heuristic

The first heuristic employs *minimum spanning trees* (MSTs) [158]. After calculating a minimum spanning tree on the data, with respect to the given distance measure, it returns the length of the largest edge in the tree. This ensures that the data set has a single connected component—which is a useful constraint in the context of cluster analysis, because we do not want to subdivide a single cluster even further. This heuristic is very efficient and suitable for enumerating 0-dimensional persistent homology. Since minimum spanning trees are very

Algorithm 11: Scale estimation based on k nearest neighbours

Require: Data set $\mathbb{X} = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$
1: **function** EstimateScaleKNN($\mathbb{X}$, k)
2: **for** Point $x \in \mathbb{X}$ **do**
3: $\mathcal{N} \leftarrow k$ nearest neighbours of x
4: $\text{dist}_x \leftarrow \max_{N \in \mathcal{N}} \text{dist}(x, N)$
5: **end for**
6: **return** $\frac{1}{n} \sum_{x \in \mathbb{X}} \text{dist}_x$
7: **end function**

sparse, it tends to be unable to find higher-dimensional topological features, except in the most simplest of data sets (see below for a comparison). Algorithm 10 contains a pseudo-code description of the heuristic.

Nearest neighbour scale estimates

The second heuristic uses scale estimates by querying the k nearest neighbours of points. The basic idea of this heuristic is to use the mean of the maximal distances in a neighbourhood as an estimate of the scale of the data. Since its calculation involves the enumeration of neighbourhoods, it is less efficient than the other heuristic. Furthermore, it may not scale well to high-dimensional data, where the concept of a 'neighbour' starts to lose its meaning and requires a more complex treatment [2, 198]. It has the advantage of being able to yield suitable thresholds for data sets with a more complex shape, though. In addition, this heuristic is more robust in the presence of outliers. See Algorithm 11 for a pseudo-code description.

Performance of estimators

To evaluate the performance of the two estimators, we check whether they are capable of recovering the correct Betti numbers of topological objects. Following the analysis of de Silva and Carlsson [337] for witness complexes, any sampling procedure should be capable of correctly recovering the topology of a two-dimensional sphere. We furthermore evaluate the estimators on a two-dimensional torus because its geometry is more complex than that of a sphere but still permits efficient sampling approaches. Table 5.1 shows a comparison of the different scale estimates. As expected, the heuristic based on MSTs does not perform well for higher-dimensional topological features, except in very simple cases where holes occur at very large scales (measured with respect to the remaining points). Although it would be possible to relax the definition of the MST and include additional edges, as suggested by Jänicke et al. [214], it is unclear how to obtain a stable threshold parameter. By contrast, the second heuristic performs very well—using less than 10% of the data points to estimate scales, we are

Object	Success rate
One circle	100%
Nested circles	100%
2-torus	0%
2-sphere	0%

MST heuristic

Object	Success rate	Neighbours used
One circle	100%	1%
Nested circles	100%	1%
2-torus	100%	6%
2-sphere	100%	7%

Nearest neighbour scale heuristic

Table 5.1: Scale estimates comparison. We used 1,000 random samples with 500 points each for each object. The table reports the highest success rate that we may achieve with the lowest amount of neighbours. A successful run means that the persistent homology calculation is able to recover the correct Betti numbers of the object.

able to recover the topological profiles of all objects. This is the heuristic we shall be using from now on.

The first heuristic still has its uses, though. We can employ it in all cases where the occurrence of higher-dimensional topological features does not increase the amount of information. For example, when analysing dimensionality reduction methods and their embeddings in Chapter 7, we are only interested in 0-dimensional persistent homology for efficiency reasons. Consequently, we do not require any higher-dimensional topological information.

5.5 Application to synthetic test data

As a more detailed analysis of the clustering algorithm, we shall briefly examine a synthetic data set of high dimensionality. To this end, we sample points from a circle, two nested circles, and a torus. We then embed these samples in a high-dimensional space. In the continuous setting, the Betti numbers—see Definition 3.15 on p. 33—of these objects would be sufficient to tell them apart. More precisely, we would have the tuples (1,1,0) for the circle, (1,2,0) for the nested circles, and (1,2,1) for the torus. In the discrete setting, in particular when high-dimensional spaces are involved, standard visualization approaches fail in identifying the differences between the objects. At this point, we may use *topological signatures* in the form of persistence rings to tell the clusters apart. Being based on the topology of the data, the persistence rings automatically detect that non-trivial topological activity only occurs in low dimensions—the remaining dimensions only give rise to a small amount of topological noise.

Creating & embedding the test data

Starting with a three-dimensional uniform grid $[a, b] \times [a, b] \times [a, b] \in \mathbb{R}^3$, we place the centre of each object at random. We then perform a proper embedding of each object into a d-

Algorithm 12: Generating a random element from $\mathrm{SO}(n)$

Require: Embedding dimension d

$R \leftarrow \left((-1)^{d-1}\right)$ ▷ Initialize a 1×1 matrix
for Every dimension k in $\{0, \ldots, d-1\}$ **do**
  $v \leftarrow$ A normalized random vector from $\mathbb{R}^k$
  $H \leftarrow I_k - 2vv^T$ ▷ Calculate Householder transformation
  $T \leftarrow \begin{pmatrix} R & 0 \\ 0 & 1 \end{pmatrix}$ ▷ Extend R
  $R \leftarrow HT$ ▷ Apply Householder transformation
end for
return R

dimensional space, with d being chosen uniformly from $[50, 100]$ using an algorithm inspired by the *subgroup algorithm* of Diaconis and Shashahani [129]. We generate a random element of $\mathrm{SO}(n)$, the *special orthogonal group* [11, Chapter 8] of $n \times n$ matrices, i.e. a random rotation within a high-dimensional space. This approach is superior to standard methods in which high-dimensional coordinates are merely obtained by random perturbations of object coordinates. Our algorithm is an isometry in the sense of Definition 4.30 on p. 75. Consequently, it does not change the intrinsic structure of an object. The basic idea of our algorithm involves generating a random rotation by means of repeated *Householder transformations* [204]. In each iteration of the algorithm, we add one additional rotation around a random vector. Hence, after d iterations, we obtain a random rotation into d-dimensional space. Algorithm 12 describes the procedure in more detail. After padding the coordinates of all objects with zeroes, we apply the rotation matrix to all coordinates and obtain a high-dimensional point cloud. This sort of noise in the data is more in line with the manifold hypothesis. In order to simulate real-world data set, we also add Gaussian noise with $\mu = 0$ and $\sigma = r$, around the positions of each object, where r is the mean inter-point distance of an object. This ensures that noise aggregates around an object and is not merely clutter in the high-dimensional space. In addition to this procedure, we add random positions to the three-dimensional grid prior to embedding the objects. This makes the objects 'harder to find', i.e. the changes in density are not as discontinuous as if the data space only contained the sampled objects.

Results & parameter stability

Using the automated parameter selection for the Rips graph as well as 10% of the data points for the density estimations, the algorithm is capable of perfectly detecting and segmenting the objects, even for larger amounts of noise. Figure 5.7 shows some typical topological signatures that appear. The glyphs make it easy to discern different clusters from each other.

Concerning the selection of the *distance to a measure* density estimator parameters, we already saw that selecting k to be of the order of approximately 10% of the cardinality of the

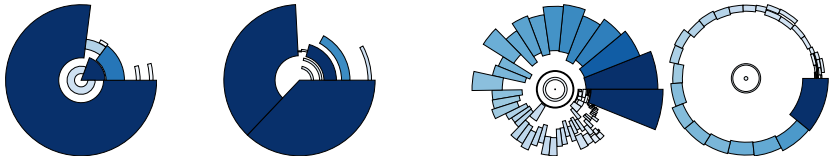


Figure 5.7: Topological signatures for the synthetic test data set. Even without using higher-dimensional persistent homology groups, the signatures are highly discriminative. The torus is the only object that contains two-dimensional topological features.

data yields sufficiently good results, even in the presence of much noise. This choice is further justified for the synthetic test data. Figure 5.8 shows that the distributions of density values maintain their bimodal shape, even if k is varied. This is a somewhat pleasant surprise because approaches using the k nearest neighbours tend to exhibit instabilities, for example when estimating gradients [115].

5.6 Application to cultural heritage data

In the following, we will see how persistence rings and persistence-based clustering can be employed to improve a classification workflow in cultural heritage. The basis of our investigation are cuneiform tablets, such as the one depicted in Figure 5.9 on p. 101. Cuneiform script is one of the earliest alphabets developed by mankind [393]. It was customary to inscribe cuneiform characters on clay tablets using a stylus made from a sharpened reed. The wedge-shaped inscriptions gave cuneiform its name—*cuneus* is the Latin word for 'wedge'. After inscribing the characters, the tablets were baked. Their creation process makes them comparatively robust, so assyriologists are even today discovering more and more of these artefacts. Most of the tablets are damaged in several ways, making it necessary to preserve them digitally. Since their transliteration requires skilled experts, of which there are few, an increasing amount of research concentrates on obtaining methods for creating automated transcriptions.

✦

In the past, most of the preservation and transcription attempts used either photography or 2D flatbed scanners. This process turned out to be very error-prone, requiring constant supervision and manual post-processing. The recent years have seen an increased prevalence of 3D scanners that greatly simplify transcription approaches. A 3D scanning device permits researchers to create very accurate digitized versions of cuneiform tablets, mostly in the form of digital meshes. These meshes contain a wealth of both geometrical and topological

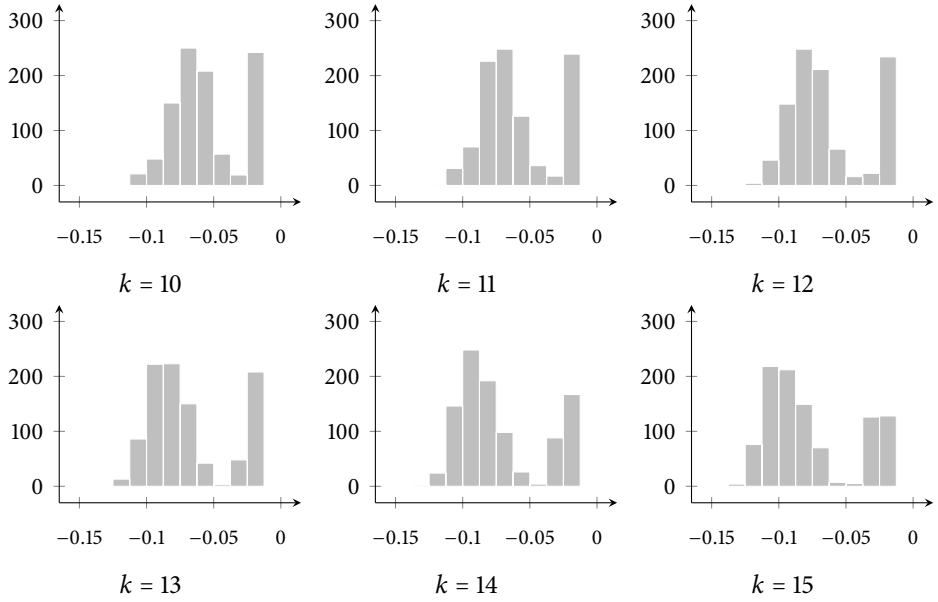


Figure 5.8: Stability of the density estimator. Slight changes in the k parameter only have a small influence on the overall structure of the density estimates. The bimodal nature of the density distribution is maintained. We used the Freedman–Diaconis rule [168] to select the bin size of the histograms.

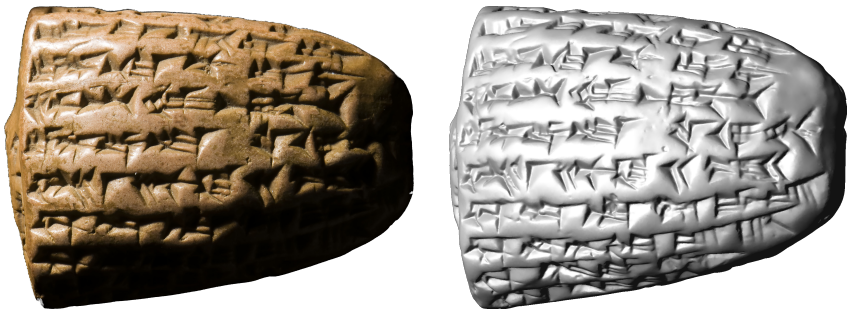


Figure 5.9: A cuneiform tablet (left) and its digitized model (right). The dimensions of the tablet are 6.2 cm × 4.6 cm × 2.9 cm, resulting in a mesh with 344,694 vertices and 689,384 faces.

information, making them useful for further analysis. Our work builds on methods introduced by Mara et al. [259, 260] who developed a multi-scale curvature filter for cuneiform meshes. The filter is based on the multi-scale integral invariant (MSII) filter introduced by Pottmann et al. [298] for the purpose of feature detection in 2D manifolds. We give a brief account of the filters in order to explain how we obtained the data for the subsequent analysis.

5.6.1 Multi-scale integral invariant filters

The *curvature* of a manifold is a fundamental concept in differential geometry [38, 236]. Intuitively, curvature measures how different a manifold 'behaves' from a flat plane. This notion of curvature is referred to as *intrinsic curvature* because it does not depend on the embedding space of the manifold. The curvature has many applications in shape matching or feature extraction. If an object contains sharp edges, for example, these can be detected by discontinuous variations in curvature. As a consequence, there is a need for robust curvature estimation for discrete data sets. For meshed data, numerous approaches already exist [5, 182, 269]. In case the mesh is defective or missing some of its parts—as is the case for cuneiform tablets acquired with 3D scanners—most methods fail to provide stable estimates and require auxiliary constructions such as Voronoi diagrams [268]. The approach by Pottmann et al. [298] instead applies a multi-scale estimation of curvature, which can be shown to be highly robust [399].

The curvature estimation at any point of the mesh requires the evaluation of the *volume integral invariant* $V_r(\cdot)$. Given a point $p \in \mathbb{R}^3$ of the input data, $V_r(p)$ is defined as the integral of the indicator function $\mathbb{1}_{\mathbb{D}}(\cdot)$ of the mesh domain $\mathbb{D}$, evaluated within a Euclidean ball B of radius r around the query point p:

$$V_r(p) := \int_{p+r\mathrm{B}} \mathbb{1}_{\mathbb{D}}(x)\,\mathrm{d}x \tag{5.4}$$

$V_r(p)$ is then normalized to a range of $[-1, 1]$. The idea of the volume integral invariant is to measure how much of a measure is inside or outside a given Euclidean ball on a certain scale. If the larger part of the volume does not contain any points of the mesh, p is locally convex. By contrast, if the larger part of the volume contains mesh points, p is locally concave. Since local convexity/concavity and curvature are intricately related, this permits us to obtain curvature estimates. To this end, we define a scale of decreasing radii $r_1 > r_2 > \cdots > r_{k-1} > r_k$ and calculate $V_r(p)$ for each radius. The result is a feature vector in some $\mathbb{R}^k$, with $8 \leq k \leq 16$ typically. By assigning each feature vector to the query point, i.e.

$$f_p := \big(V_{r_1}(p), \ldots, V_{r_k}(p)\big), \tag{5.5}$$

we assign the input mesh its feature space. Pottmann et al. [298] prove that the feature space may be used to obtain accurate approximations for the local convexity or concavity of the input mesh. By doing the calculations at multiple scales, the estimates are robust against noise.

Properties of the feature vector space

Prior to analysing the space of feature vectors, we briefly elucidate its topological structure. We first note that, in the continuous case, the k-dimensional feature vector calculation is a function $f\colon \mathcal{M} \subseteq \mathbb{R}^3 \to \mathbb{R}^k$ from a manifold $\mathcal{M}$ embedded in three-dimensional space to a k-dimensional real space. Common 3D scanning devices result in *meshes*—essentially simplicial complexes—that are triangle-based. Hence, the maximum intrinsic dimension of $\mathcal{M}$ is 2. We can also characterize the image of f.

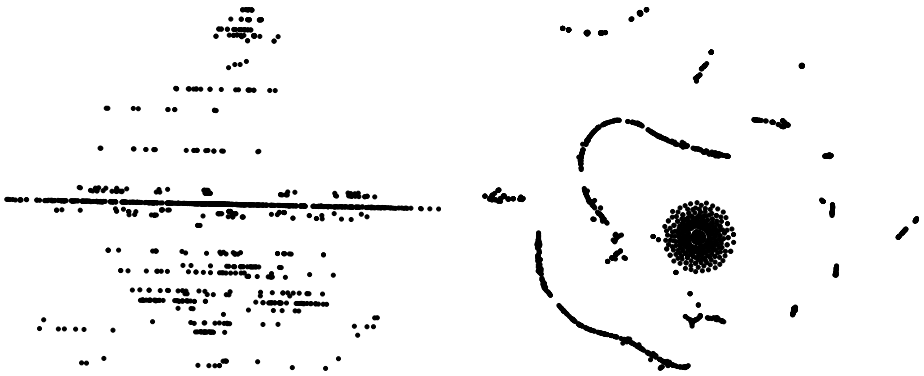
Theorem 5.1 (Feature vector space dimension). The intrinsic dimension of the image $\operatorname{im} f := \{f(x) \mid x \in \mathcal{M}\} \subseteq \mathbb{R}^k$ is at most 2.

Proof. Clearly, f is a smooth map between two manifolds. As a consequence, its *rank* is well-defined [238, pp. 77–78]. By the *rank–nullity theorem* [11, pp. 110–111], the dimension of $\operatorname{im} f$ cannot be larger than the dimension of $\mathcal{M}$, and the claim follows. ∎

Intuitively, this makes sense—if we calculate any continuous function on some manifold, we cannot expect the results of this calculation to suddenly attain a higher dimensionality than the original space. Knowing that the feature vector space is a manifold on its own makes it an interesting subject for topological analysis. In particular, the previous theorem implies that we only need to calculate persistent homology up to dimension 2—there cannot be any higher-dimensional topological features.

Embedding challenges

Despite our knowledge about the intrinsic dimension of the space of feature vectors, they turn out to be very challenging to analyse. Dimensionality reduction algorithms, such as MDS [49] or t-SNE [256], tend to be confused by the correlations in the data. Since $V_r(p)$ changes continuously when slightly changing the radius, its values will be highly-correlated along different scales. This results in either very linear structures, as shown in Figure 5.10a, or a misrepresentation of local neighbourhoods, as depicted by Figure 5.10b. The latter embedding suggests a clustering structure that is not present in the data. These issues are not mitigated by applying standard data pre-processing operations and decorrelation techniques, such as *centring* and *sphering* [111].



(a) MDS (b) t-SNE

Figure 5.10: Example embeddings of synthetic MSII data. Dimensionality reduction algorithms either focus too much on linear structures in the data (left) or misrepresent local neighbourhoods (right). Hence, standard dimensionality reduction methods cannot be used to analyse MSII feature vectors.

Visualization challenges

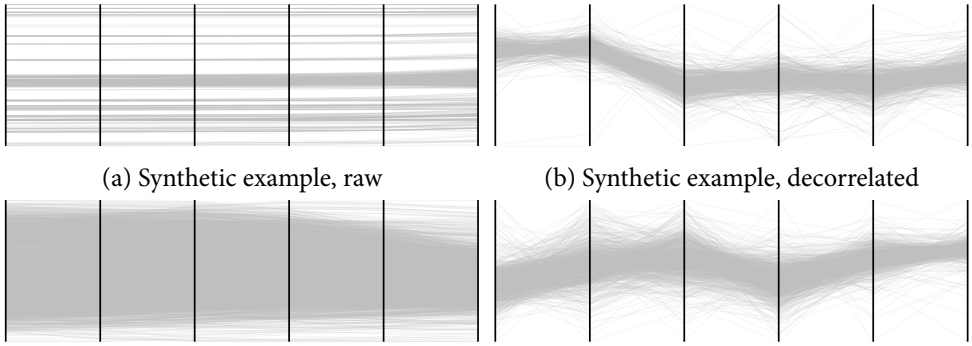
When applying standard data visualization techniques to the data, patterns start to emerge. Figure 5.11a shows *parallel coordinate plots* (PCPs) [210] of a synthetic MSII data set. They serve to display the grouping behaviour at different scales. With decorrelated data, however, parallel coordinates start to become very cluttered, as depicted by Figure 5.11b. Moreover, MSII filter responses for real-world data sets do not exhibit any easily-discernible structures in the parallel coordinate visualization (Figures 5.11c and 5.11d). By contrast, persistent homology has a built-in way of dealing with multi-scale aspects of a data set. MSII filters are thus a prime candidate for topological data analysis.

✦

In the following, we shall analyse the topology of a 16-dimensional MSII feature vector space. Our analysis has three goals. First, we want to find out whether single cuneiform characters may be extracted from a mesh solely by their topological information. Second, we want to investigate qualitative differences between the feature spaces of different cuneiform characters. Third, we want to confirm whether the scales of the MSII filter have been selected correctly. We will see that our proposed workflow based on clustering and topological signatures, given by the persistence rings, is capable of reaching these goals.

Previous work

Mara [259] developed a technique for extracting cuneiform characters. The method involves the convolution of feature vectors and relies heavily on thresholding. While Mara demonstrates



(a) Synthetic example, raw (b) Synthetic example, decorrelated

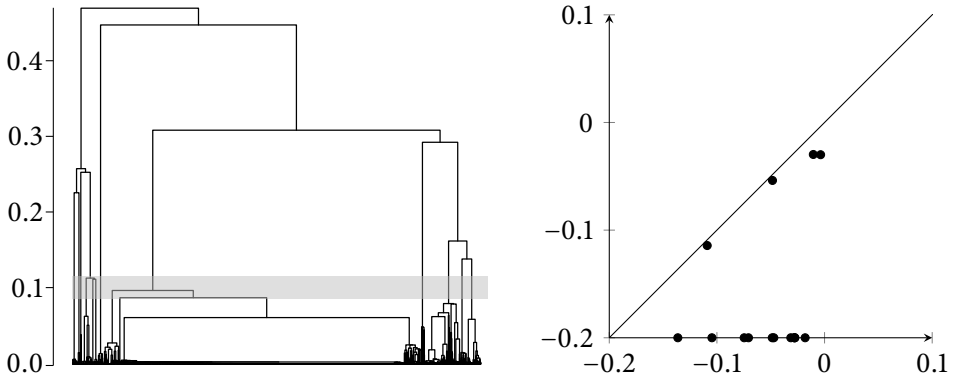
(c) Real example, raw (d) Real example, decorrelated

Figure 5.11: Parallel coordinate plots of typical MSII filter responses. For reasons of clarity, the figures only show the six most expressive radii of the filter. We can see that different parts of a data set result in clear filter response profiles, but only in synthetic data (top row). In real-world data (bottom row), the responses contain more noise, leading to cluttered visualizations. As the visualizations on the right-hand show, standard decorrelation operations destroy salient structures.

very good results using a priori knowledge, a completely automated approach is still not feasible for several reasons. First, the feature vector calculations requires numerous radii; incorrect choices will result in over- or under-sampling features in the space, leading to error-prone character extractions. Furthermore, the threshold cannot be easily chosen and the character extraction is unstable with respect to small changes in the threshold value. Last, the result of the extraction process cannot be validated automatically yet—hence still requiring tedious comparisons between the digitized tablets and the extraction results.

Our approach

With these issues in mind, we shall attempt to resolve them through topological data analysis. The following analysis will solely use the high-dimensional feature space of the mesh. We will not make use of any connectivity or positional information stored in the input mesh. Our approach consists of two larger steps: To solve the extraction problem, we shall use the persistence-based clustering algorithm first, resulting in a partition of the input space into smaller clusters. We shall see that these clusters turn out to describe different regions in the corresponding cuneiform tablet. Having obtained the clustering information, we shall calculate the topological signature of each cluster. We will see how this signature helps us to decide whether a cluster contains any meaningful structures. Topological signatures thus turn out to be a first step towards an eventual automated validation of character extraction algorithms. Our analysis will also uncover instabilities in the calculation of individual feature calculations, thereby warranting further research into the stability of integral invariants.



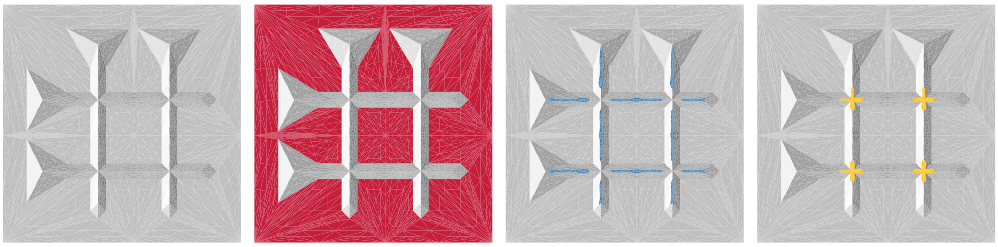
(a) Dendrogram (b) Persistence diagram

Figure 5.12: Dendrogram and persistence diagram for the 'Kaskal' data set. Most distance variability is encountered within a scale of $[0.075, 0.125]$. For larger values, the data set decomposes into clusters that are too coarse. The clusters show up as clearly-separated points in the persistence diagram. This is due to the synthetic origin of the data.

Through interacting with persistence rings, we will discover a hitherto unknown complicated nested relationship between several classes of feature vectors. Following the discussion from above, in particular the statement from Theorem 5.1, we will calculate persistent homology until dimension 2 only.

5.6.2 Synthetic data set

We first analyse a synthetic mesh depicting the cuneiform character 'Kaskal' 𒍪. The 'Kaskal' cuneiform character, meaning 'way', 'travel', or 'expedition', most likely has a non-Sumerian origin. Its very regular structure with multiple wedges and intersections makes it a useful training character. The corresponding mesh does not contain any noise due to its synthetic origin, thereby ensuring that its planar parts are executed perfectly. Hence, there is no difference in the height of the ridges which we would expect when a human writes the same character. Our automated parameter selection algorithm suggests to use $\epsilon \approx 0.1$ for the subsequent analysis. Figure 5.12a shows a dendrogram of the data, which indicates that the cluster variability appears to be sufficiently good for $\epsilon \in [0.075, 0.125]$. Larger values for ϵ will result in few large clusters, while smaller values yield only a large number of unstable clusters. For this parameter range, we obtain the persistence diagram depicted in Figure 5.12b. The individual clusters manifest themselves as points on the abscissa of the persistence diagram.



Complete mesh | Planar parts | Bottom of wedges | Intersections of wedges

Figure 5.13: The clusters detected by the clustering algorithm mapped to the original input mesh. Note how the different parts of the ridges are decomposed into multiple clusters.

Clusters

The clustering algorithm resulted in twelve clusters, of which we only show a selected few in Figure 5.13. Each cluster describes a set of feature vectors that describe different regions within the input mesh. We first observe that the red cluster ●, which contains the largest amount of feature vectors, describes the planar parts of the mesh. Since all planar parts of the mesh 'behave' exactly the same, meaning there are no oscillations in planarity, their feature vectors will form a very dense region in the parameter space. This dense region is easily detected by the clustering algorithm. Note that also some parts of the V-shaped wedges are assigned to this cluster because the planarity decreases continuously. Another easy-to-explain structure is described by the blue cluster ● that contains all points of the bottom of each ridge. All these points are part of extremely concave regions in the mesh and their feature vectors again form a dense region in the parameter space. The remaining clusters contain feature vectors from different interesting substructures in the mesh—the yellow cluster ●, for example, is formed by the feature vectors that correspond to points where wedges intersect.

Persistence rings

We then proceed to calculate several persistence rings of some representative clusters in the data set. Figure 5.14 shows the results. Even by a cursory visual inspection, differences in the 1-dimensional persistent homology of the clusters are apparent. Thus, users may exploit this information without having to rely on other statistics such as cluster size. When analysing the relative position of clusters to each other within the feature space, we found out that the largest cluster, i.e. the one whose feature vectors describe planar regions, is part of the boundary of the other clusters. In contrast, the other clusters only feature two or three holes of larger persistence—and the corresponding feature vectors merely bound smaller parts of the feature space. These findings indicate a complex nested relationship between different feature vectors that warrants further study. Hitherto, the relations between parts of the feature vector space

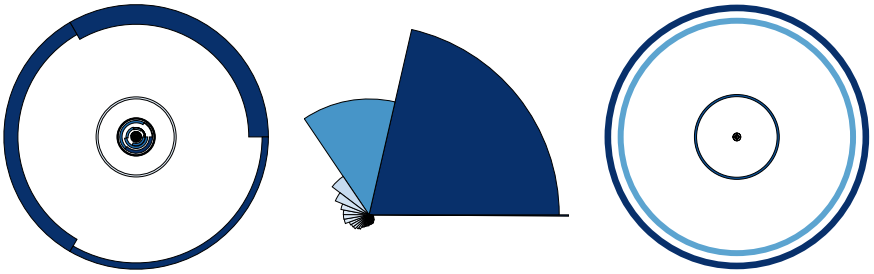


Figure 5.14: 1-dimensional persistence rings for representative clusters of the 'Kaskal' data set. Both the amount of holes and the distribution of persistence values visually differs in all partitions. All features are well-separated with respect to their scales because of the synthetic nature of the data.

have not been exploited. It is reasonable to assume that they could be used to improve the quality of the threshold-based segmentation.

5.6.3 Real data set

As a real-world example for the application of our workflow, we used a digitized cuneiform tablet from the Heidelberg Institute of Assyriology. Internally, the tablet is denoted 'G8'. With 344,694 vertices and 689,384 faces, its resolution is sufficiently good to even retain the most intricate of details such as fingerprints. At the same time, this resolution makes automated character extraction very difficult. Our initial analysis showed that the feature vectors are very noisy. There are two different noise sources. One comes from the instabilities in the calculation of feature vectors themselves, the other is caused by cuneiform characters that are imperfectly imprinted on the mesh. The amount of noise explains why thresholding techniques are prone to instabilities. In order to make the data less noisy, we will be working with several regions of interest in the mesh instead of the complete mesh. This also reduced the running time of our algorithm from 300 s to a mere 5 s. Even with data reduction, dendrograms turned out to be impractical for choosing a distance threshold. We thus relied on the automated parameter selection heuristics described in Section 5.4.

Properties of the clustering

The persistence diagrams that are obtained using this sampling procedure turn out to exhibit a clear distinction between topological noise and topological signal. This is an indicator of the robustness of our approach and confirms the utility of using topological methods in this context. Figure 5.15 shows the persistence diagrams of all regions of interest. A closer investigation of the clustering results shows that upon increasing the ϵ parameter, the majority

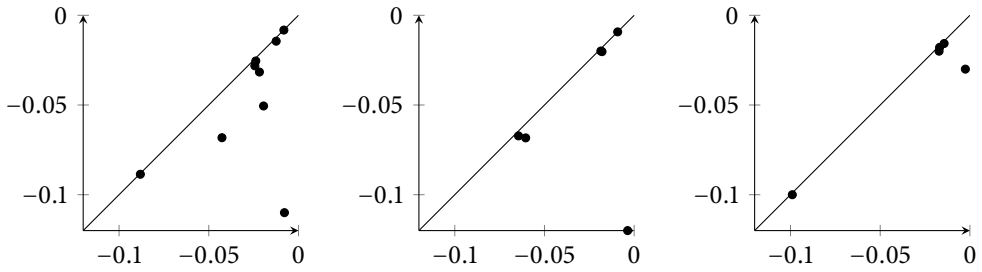


Figure 5.15: Persistence diagrams of regions in the 'G8' data set. The persistence diagram of each region contains a small amount of topological noise that is clustered around the diagonal. There is a clear distinction between noise and signal.

of the data points will be merged into a single large cluster. This is caused by a skewed density distribution within the feature vector space. Every meshed object contains a large amount of 'regular' regions, i.e. regions that are fully planar or almost planar. The corresponding feature vectors will thus share a similar profile, resulting in a very dense region. Non-planar regions in the mesh, by contrast, will result in very different feature vectors—in particular, those feature vectors will be very different from each other, resulting in a myriad of small clusters that contain only few vertices (in comparison to the single cluster of regular points). Upon increasing ϵ, these small clusters will rather be merged into the densest cluster than into each other. To prevent further skewing the density distribution, we merged clusters into the nearest cluster with respect to the average linkage distance. This prevents the chaining effect that often occurs in single-linkage clustering [189]. The merge process destroys very fine distinctions in the mesh but ensures that the clustering is very stable against noise. The resulting clusters describe a segmentation of the mesh in which the 'skeleton' of each cuneiform character is extracted. Figure 5.16 depicts examples of the resulting segmentation.

We have to re-iterate again that the algorithm did not use any of the connectivity information provided by the mesh vertices and edges. Rather, only attribute space information in the form of the MSII feature vectors was used. This shows that the topological information carried by the feature vectors is sufficient to discriminate between different regions in a cuneiform character mesh.

Persistence rings

After the merge process, we calculate topological signatures in the form of persistence rings for each of the clusters. Figure 5.17 depicts the results. The signatures agree with our analysis of the synthetic cuneiform data. When comparing the 'background' clusters of each region of interest, we observe that they exhibit the same topology regardless of their cardinality, i.e. the number of data points they are made up of. Topological features in this cluster have many

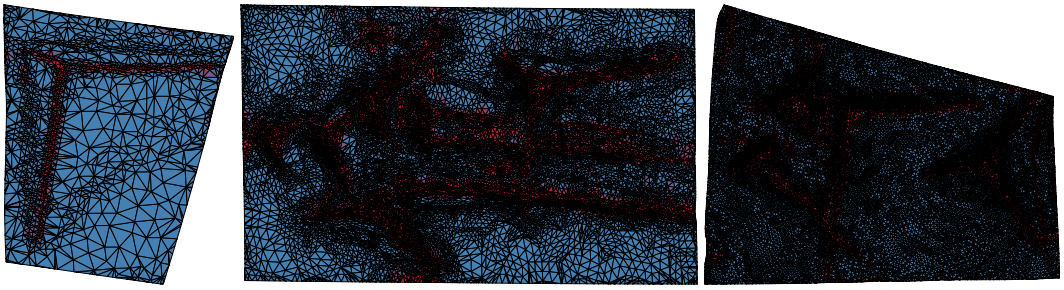
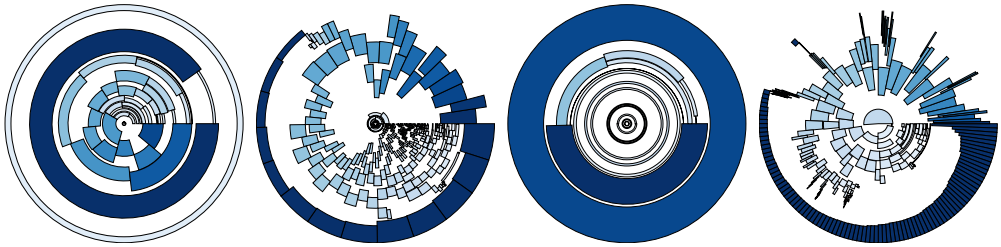


Figure 5.16: Cuneiform skeletons of three regions of interest for the 'G8' data set. The skeletons have been extracted by means of topological information alone, which proves that the topology of the feature vectors is a salient descriptor of the data.

overlapping scales, leading to a very distinctive appearance of the persistence ring. This fits our description of the feature vector space. The space of feature vectors contains numerous holes that bound smaller clusters, which correspond to different parts of the cuneiform character.

By contrast, the 'character' cluster, i.e. the data points that contain the cuneiform character skeleton, do not exhibit as many 1-dimensional topological features. Furthermore, their persistence rings contain less topological features whose scales overlap. This is indicated by the larger angles that occur in the annular sectors. We thus have an easily-recognized property for visually distinguishing the two clusters on each region of interest. By solely using the persistence rings, this distinction does not require any complex visualizations of the feature vectors themselves. Just as for the 'Kaskal' data set, the persistence rings hint at a nested relationship. Due to the large amount of noise in the feature vectors, the topological structure of the 'character' clusters loses some details. If the radii for calculating feature vectors were varied based on the local density of the mesh, the results would be less unstable, which in turn would permit us to learn more about the topological properties of the space.

In summary, the persistence rings of these regions of interest matches the previously-encountered phenomenon. The feature spaces again contain a large amount of holes that bound the smaller clusters corresponding to the cuneiform character skeletons. As a consequence, the persistence rings of the 'character' clusters contain a smaller amount of 1-dimensional generators than the 'background' clusters. The topological structure of the 'background' clusters is richer because they contain more feature vectors with slightly different scale behaviour—whereas the 'character' clusters only contain those feature vectors that are almost similar on all scales. It is possible to use the amount of topological noise, i.e. features of small persistence with slightly-overlapping scales, as an indicator of the amount of noise that underlies the feature vector calculation.



'Character' cluster 'Background' cluster 'Character' cluster 'Background' cluster

Figure 5.17: 1-dimensional persistence rings for some 'G8' regions of interest. Even though the clusters have different cardinalities, the general shape of the persistence rings remains the same.

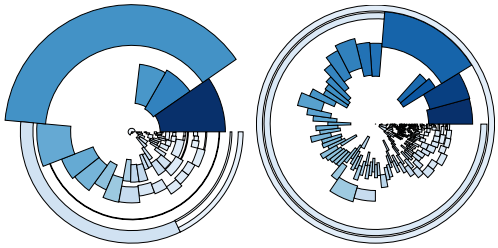
Detecting unsuitable radii

Having seen that the persistence rings yield a salient visual description of the MSII feature vectors, we now take a look at what happens when we choose unsuitable radii during their construction. If the filter radii are too large, they will result in 'washing out' structures in the data. The differences between sharp edges or wedges of the cuneiform characters will not be evident any more. The selection of suitable radii for the MSII feature vector calculation is still tedious and requires manual inspection. Mara et al. [260] suggest setting the largest radius to be slightly larger than the largest wedge on a cuneiform tablet.

Our topology-based approach permits the immediate detection of unsuitable radii by comparing persistence ring visualizations to the expected results. Figure 5.18, for example, shows two persistence rings that have been calculated with a radius that is too large. Here, the 'background' cluster does not exhibit a similar amount of high-persistent topological features, for example. Likewise, the 'character' cluster exhibits more topological features whose scales overlap. We do not observe this behaviour for suitable radii, such as the ones shown in Figure 5.17. Persistent homology thus yields a visual criterion for determining the suitability of MSII feature spaces without requiring complicated threshold selections. The advantage of persistence rings in this context is that they do not require the complete feature vector space in order to yield a structural description of the data. Hence, multiple threshold selections can be tested by sampling the data and calculating persistence rings on each sample. Suitable thresholds can then be detected visually, saving gratuitous calculations.

5.7 Discussion

This chapter introduced a novel workflow for multivariate data analysis, combining topology-based clustering algorithms with the extraction of topological features for each cluster. The topological features are displayed using a new visualization, the *persistence rings*. We saw how



'Character' cluster 'Background' cluster

Figure 5.18: 1-dimensional persistence rings for unsuitable radii. The 'character' cluster, for example, contains many features whose scales overlap. This behaviour is markedly different from the persistence rings for suitable radii.

to combine this efficient and compact visualization with cluster analysis, yielding a set of *topological fingerprints*.

We then employed this new method for analysing multivariate data sets to explore the topological structure of feature vector data. Our method helped understand the phenomenon of noise in the data and helped explain why previously-considered threshold approaches for cuneiform character extraction are unstable. In addition, we were able to segment the feature vector space into numerous clusters, each corresponding to a different part of a cuneiform character. This segmentation process used solely topological information. Building on that, we observed that the topological signatures of the clusters, visualized through persistence rings, had clearly-defined differences, making it possible to differentiate individual parts of cuneiform characters in several regions of interest on a cuneiform tablet.

Extensions: Discriminative properties

A large swath of work lies ahead concerning the applicability of persistence rings for distinguishing between different data sets. The efficacy of this visualization depends on suitably-defined distance measures on the data set as well as on useful approximations of the connectivity structure of the data set, given in the form of Rips graphs. A natural extension would be to employ *metric learning* [122, 228] in order to obtain useful distance measures for the MSII feature vectors. Measures based on the *Mahalanobis distance* [125, 258] may also result in improved segmentations.

Future work also should explore the potential of other geometric graphs for data analysis. In the opinion of the author, the integral part played by a neighbourhood graph has been largely ignored so far. The Rips graph, the Čech complex, and the Vietoris–Rips complex are commonly used because their homotopy approximation properties are known. In the context of scalar field visualization, Correa and Lindstrom [115] show that modifications of standard neighbourhood graphs are required, particularly when the sampling is sparse. From

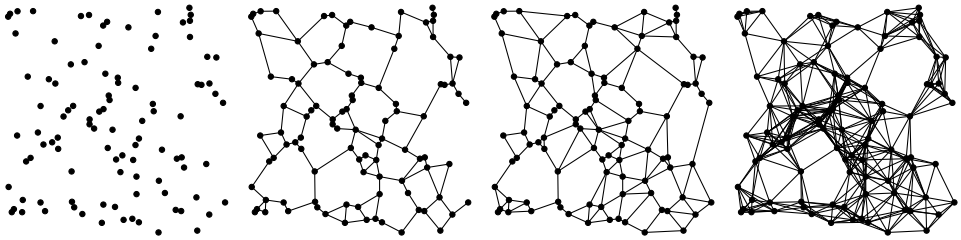


Figure 5.19: Some geometric graphs of a randomly-sampled unit square. The β-skeleton, with $\beta = 1.5$, is slightly sparser than the Gabriel graph. The Rips graph $\mathcal{R}_\epsilon$, with $\epsilon = 0.2$, treats all neighbours equally, which results in a rather dense graph.

the point of algebraic topology, these neighbourhood graphs do not preserve the homotopy type of the data. Moreover, their expansion in the form of a flag complex is unlikely to yield any useful information, except for persistent homology in dimension 0. There are further *geometric graphs*, which can be used to extract connectivity information from data. The spectrum ranges from simple tree-based structures, such as more robust minimum spanning trees [403], to subgraphs of the *Delaunay triangulation* [364]. While proximity graphs such as the *Gabriel graph* [173] are known to introduce spurious connections between data points, in general the stability of these structures seems to be useful for topological data analysis and visualization—in particular, the absence of a threshold parameter makes creating these graphs easier. Figure 5.19 depicts several geometric graphs of random samples from the unit square. The author considers a combination of non-parametric geometric graphs with diffusion processes, random walks, or Markov chains to be a useful model. Bendich et al. [35] already proved that random walks have the potential to further improve the stability of persistent homology calculations under certain conditions, making this technique a very promising approach. The largest issue to tackle is the difference in scales that is inherent to real-world data. As a first step, the author proposes the development of a Rips graph whose local scale is permitted to vary. Diffusion processes could provide a smoothing of local scales so that there are no abrupt changes in homotopy type.

Extensions: Visualization

The persistence rings themselves could be improved by exploiting ideas found in hyperbolic visualizations. They are demonstratively able to guide the focus of users towards relevant objects and have been successfully employed in the visualization of hierarchies [231]. Subsequent work, in particular by Munzner [278], has shown the utility of these visualizations for graph drawing. For persistent homology, it could be used to direct the attention of viewers

towards interesting topological features—maybe in conjunction with improved criteria for their significance.

If persistence rings are calculated for larger data sets, random sampling approaches could be used to obtain typical topological signatures. In this case, an improved layout heuristic should take the *uncertainty* of a topological feature into account, for example by shading or scaling the annular sectors accordingly. Such a sampling approach could make use of *witness complexes* [337], especially as the amount of feature vectors increases. The bottleneck of these calculations, however, is not the persistence ring visualization on its own but rather the general persistent homology pipeline.

6 Structural analysis of point clouds using simplicial chains

In the preceding chapter, we have seen how to use topological methods as a 'fingerprint' technique that describes data via auxiliary visualizations such as persistence rings. Another facet of multivariate data sets involves their intrinsic geometry, as measured by pairwise distances, for example. Persistent homology only includes little information about the geometry underlying a data set. In this chapter, we will thus augment persistent homology with geometrical information from multivariate data sets. We will describe a novel algorithm that aims to calculate concise topological descriptions in a data set. These descriptions will be visualized using the *simplicial chain graph*, a new visualization that combines both geometrical and topological aspects of a data set. We will use the simplicial chain graph to analyse two complex data sets that vary over time. The contents of this chapter are based on a previous publication [316] by the author.



The methods presented in this chapter are meant to support the analysis of data sets that have been sampled at snapshots in time. Interpolation is not always easily possible, in particular for complex multivariate data sets. For instance, when analysing the voting data in Section 6.4.1, different voting periods cannot be interpolated in a well-defined manner. Nevertheless, as a multivariate data set undergoes transformations—not necessarily continuous ones—both its topology and its geometry change. Since topological changes tend to happen more gradually, we want to focus on them and connect them to geometrical changes. This permits us to depict qualitative changes between time-varying multivariate data sets.

Motivating example As a motivating example, suppose we are given a set of points in $\mathbb{R}^2$ that are roughly arranged in the form of two circles. We now let the points oscillate to some extent, such that the circles continuously shrink and expand. Figure 6.1, top, on p. 116 depicts this situation. Using persistent homology, we can easily detect that the data points are situated along the boundary of two 1-dimensional holes. Instead of focusing on all of their positions, we simply focus on the properties of the holes and how their size changes during the oscillation

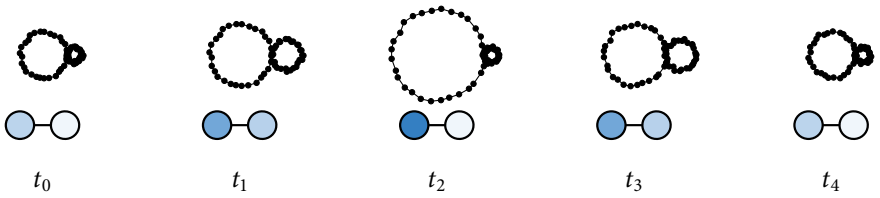


Figure 6.1: An example motivating structural analysis. The circles are an extremely simple data set whose shape changes over time. Persistent homology offers a compressed view of these changes by focusing merely on how the holes made by the circle change. This is indicated by the graph below the circles. In the graph, every node corresponds to a hole and edges connect holes that are close.

process. We collect this information in a graph, as shown in Figure 6.1, bottom, that represents both holes as nodes. Node colours correspond to the size of the hole, with darker colours indicating larger holes, while the edge connects the nodes in order to indicate that the two holes are close to each other. By observing how this graph changes at different snapshots of the data set, we get an overview of qualitative topological changes. Such a graph also yields a compressed view of the data. In this example, the graph of the middle time-step indicates somewhat erroneous behaviour—one of the holes is very large (dark blue), the other one very small (light blue). Note that this simple example serves only as an illustration; for multivariate data sets, issues like this cannot be spotted as easily.

Our approach In order to capture the changes in connectivity that are inherent to a data set, we will describe them by *simplicial chains*, i.e. sets of simplices that correspond to the boundary alongside a high-dimensional hole. Since the simplicial chains that are calculated using the standard persistent homology algorithm do not take the geometry of the data set into account, our first goal will be to provide them with more geometrical information.

6.1 Why do we need geometrical information?

We have seen in the previous chapters that algebraic topology is somewhat insensitive to the geometry of data. If we return to the core definition of a topological feature, we recall that persistent homology tells us about the existence or presence of holes in our data. However, telling us about the presence of a hole is not sufficient—ideally, we want a description in terms of our input data. Such a description is being given in the form of a *simplicial chain*, as described by Definition 3.7 on p. 30. Chan et al. [87], for example, showed how such descriptions may be used to better understand the evolution of viruses. In mathematical

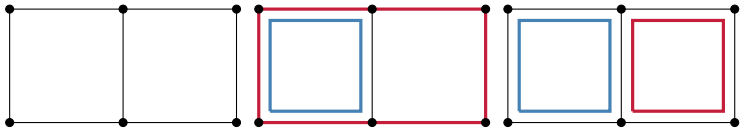


Figure 6.2: An illustration of the need for localized simplicial chains. The simplicial complex shown in (a) contains two holes of dimension 1. A valid set of generators for these holes, i.e. sets of 1-simplices, is depicted in (b). None of the generators corresponds to our intuition of a hole, though. We would prefer the set of generators as shown in (c). To obtain generators such as these, we need to have a measure of the size of a simplicial chain.

terms, we want a description of the *generator* of a hole, i.e. a set of all those simplices that are situated around the boundary of the hole.

6.1.1 The localization problem

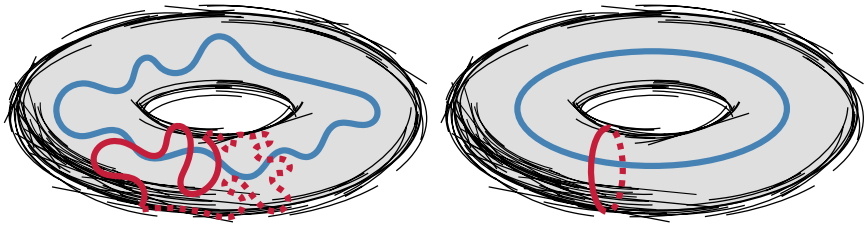
Recalling Algorithm 5 on p. 62, we obtain such a description in the form of a *cascade* of simplices. The crux of the matter, however, is that there are usually many permissible sets of simplices that may serve as generators. Following the definition of a k-dimensional hole, we are looking for any set of simplices that are neither the boundary of a $(k+1)$-dimensional simplex nor have a boundary themselves

Suppose that we have a data set consisting of two squares. Persistent homology should thus detect two 1-dimensional holes. Their *presence* is detected correctly by persistent homology, but their simplicial chains might not look as expected. Figure 6.2, inspired by an example of Zomorodian and Carlsson [410], illustrates this issue and shows that not all algebraically valid solutions coincide with our intuitive notion of a hole.

The issue gets even more problematic for data with a pronounced geometrical structure. Figure 6.3 on p. 118 shows two sets of 1-dimensional simplicial chains for a torus. Both sets serve as valid generators for the same homology classes, but one is more local than the other. The problem of finding these generators is thus named the *localization problem*. Especially in high-dimensional data sets, where we cannot readily visualize the simplicial chains, we want them to be as concise as possible in order to describe meaningful structural information.

6.1.2 A notion of conciseness

Having seen the issue, what constitutes a concise description of a hole in a simplicial complex? Ideally, from all possible descriptions of a generator, we would like to pick the one that has the fewest amount of simplices. For 1-dimensional simplicial chains, an algorithm by Erickson and Whittlesey [159] finds an optimal solution in polynomial time. Unfortunately, Chen and



(a) 'Bad' loops (b) 'Good' loops

Figure 6.3: 'Bad' and 'good' loops on a torus. Both sets of loops are valid. When thinking about the torus, algebraic topologists usually consider the loops in (a) to be less preferable than the ones in (b), although they are both equally valid. Our criteria for *simplicial chain localization* developed in this chapter will ensure that we arrive at the latter type of loops.

Friedman [94] showed that this is NP-hard to to approximate within any constant factor for higher dimensions. We thus need to relax the definition somewhat in order to obtain solutions in polynomial time. This does not necessarily imply that these solutions can be found efficiently, though.

To find a suitable relaxation, we recall that a simplicial chain is not uniquely-defined. We are allowed to add simplices as long as we do not leave the homology class of the chain. Instead of trying to figure out how to reduce the size of the simplicial chain, we may thus simply concentrate on reducing the size of the simplicial complex—more precisely, we want to find the smallest simplicial subcomplex of our data that *supports* the chain, i.e. that still contains all the simplices that are part of the chain. The difference to the previous definition may seem subtle, but while our previous definition required the simplicial chain to have as few simplices as possible, we now place no limit on its cardinality and rather measure its extents with respect to the complete simplicial complex of the data set. This is motivated by optimization strategies for real-valued problems, where one restricts certain results to Euclidean balls whose radius is as small as possible. The relaxation permits us to transform a complicated optimization problem into the language of graph optimization problems, because it will turn out that we only need the 1-skeleton, i.e. the neighbourhood graph, of the simplicial complex in order to optimize simplicial chains.

6.2 Localizing simplicial chains

Definition 3.7 on p. 30 already described a k-dimensional simplicial chain as a *formal sum* of k-simplices. Intuitively, we can think of each simplicial chain as representing a closed path without a boundary in a simplicial complex. If we take a look at the simplicial chains in in Figure 6.3, for example, we see that they describe a closed path of 1-simplices, i.e. edges. Hence, such paths constitute a hole in the simplicial complex. In this figure, we also see the

basic problem with simplicial chains: There are many closed paths of k-simplices that satisfy the definition of a k-dimensional simplicial chain. Most of them do not correspond to our intuitive notion of a hole.

Chen and Freedman [95] proposed an algorithm for obtaining a localized description of simplicial chains. It follows the ideas of a 'small simplicial complex' outlined above but does not make use of the weights in the simplicial complex. We subsequently develop an algorithm that is based on their work and show its correctness. Our algorithm has a better runtime behaviour and is more general—it works for Rips graphs with arbitrary weights. Furthermore, our experiments indicate that the inclusion of edge weights results in geometrically more meaningful simplicial chains. The algorithm can be easily implemented, is parallelizable, and works for simplicial complexes of moderate size. The set of localized simplicial chains belonging to a given data set will play a central role in devising a visualization for obtaining a structural description of a multivariate point cloud.



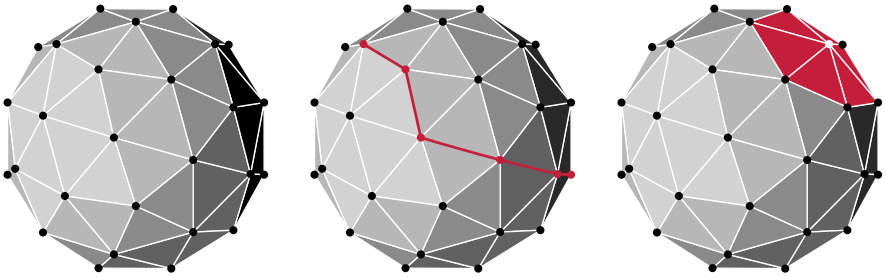
We shall subsequently assume that we want to localize a set of simplicial chains in a simplicial complex K. For technical reasons, we assume that these simplicial chains belong to different homology classes in the simplicial homology groups of K. To define 'small simplicial complexes', we first require a notion of distances in a simplicial complex.

Definition 6.1 (Geodesic distance in simplicial complexes). Given a simplicial complex K with weighted simplices, we define a distance function $\operatorname{dist}_v \colon \operatorname{vert} \mathrm{K} \to \mathbb{R}$ for any vertex v by setting $\operatorname{dist}_v(w)$ to be the weighted length of the shortest path connecting v and w, measured using the 1-skeleton of K. We can extend this function to model the distance from a vertex v to an arbitrary simplex $\sigma \in \mathrm{K}$ by setting

$$\operatorname{dist}_v(\sigma) := \max_{w \in \operatorname{vert} \sigma} \operatorname{dist}_v(w), \tag{6.1}$$

as we have done earlier when calculating a filtration of the Vietoris–Rips complex. The extended function $\operatorname{dist}_v \colon \mathrm{K} \to \mathbb{R}$ models the *geodesic distance* within the simplicial complex, i.e. the shortest distance between a vertex v and another simplex.

The term *geodesic distance* refers to the fact that the distance is measured using the 'interior' of the space. Geodesics [38, pp. 38–53] are often used in Riemannian geometry where the calculations are not performed in the ambient space but rather on a manifold, for example. Figure 6.4b shows an example geodesic of a simplicial complex corresponding to a sphere. We can use the definition of geodesic distances to obtain a valid filtration of a simplicial complex K. This is required to ensure the correctness of the subsequent algorithms.



(a) A simplicial complex (b) A geodesic (c) A geodesic ball

Figure 6.4: Geodesics for simplicial complexes. In (a), we see one possible simplicial complex that triangulates a 2-sphere. A possible geodesic between two vertices is shown by (b). In this case, the geodesic is also the shortest path connecting the vertices. Finally, we can see a geodesic ball in (c), centred around the vertex marked in white. It contains all the highlighted simplices as well as their faces.

Lemma 6.2. The ascending ordering induced by the values of $\operatorname{dist}_v(\cdot)$ results in a valid filtration of the simplicial complex K.

Proof. Given any simplex $\sigma \in K$, we have $\operatorname{dist}_v(\tau) \leq \operatorname{dist}_v(\sigma)$ for any face $\tau \subseteq \sigma$ because $\operatorname{vert} \tau \subseteq \operatorname{vert} \sigma$ and the values of $\operatorname{dist}_v(\cdot)$ can only increase when adding additional vertices. ■

The notion of geodesic distance permits us to speak of a geodesic ball on a manifold. A geodesic ball is similar to an open ball in Euclidean space. The only difference is that it never extends beyond the simplicial complex in which it is defined. Using the definition from above, we can define the combinatorial counterpart of a geodesic ball.

Definition 6.3 (Geodesic ball in a simplicial complex). A *geodesic ball* in a simplicial complex K with centre vertex v and radius r is defined as the subset of K that contains all simplices whose geodesic distance to v is less than or equal to r:

$$\mathrm{B}_v(r) := \{\sigma \in K \mid \operatorname{dist}_v(\sigma) \leq r\} \tag{6.2}$$

Figure 6.4c depicts an example of this definition. In 3D, this definition is similar to the k-neighbourhood of a vertex in a mesh. However, our definition extends to high-dimensional simplicial complexes.

We will use the previous definition to define the *geodesic radius* as a notion of the conciseness of a simplicial chain. We first observe that $\mathrm{B}_v(r)$ is always a valid simplicial complex, or, more precisely, a *simplicial subcomplex*.

Lemma 6.4. Given a simplicial complex K and any vertex $v \in \operatorname{vert} K$, the geodesic ball $\mathrm{B}_v(r)$ is a simplicial subcomplex of K for all values of r.

Proof. We know that by sorting K according to $\mathrm{dist}_v(\cdot)$, we obtain a valid filtration. Since $\mathrm{dist}_v(\cdot)$ can only increase when adding a vertex to a simplex, any subsequence of simplices from K is closed with respect to the face relationship. The subsequence thus is a simplicial subcomplex of K. ∎

Taking stock of the definitions we introduced so far, we see that we now have the means to speak of *geodesic balls* in a simplicial complex. These geodesic balls turn out to be valid simplicial subcomplexes of the original simplicial complex. Since each geodesic ball also has an associated radius—serving as a size measure—we may now define the size of a simplicial chain by using *relative simplicial homology* as described in Chapter 3, Section 3.3, p. 34 ff.

Definition 6.5 (Size of a simplicial chain). Let c be a p-dimensional simplicial chain in a simplicial complex K and

$$\mathcal{L} := \left\{ \mathrm{L}_i \mid i \in \{1, \ldots, k\}, \mathrm{L}_i \subseteq \mathrm{K} \right\} \tag{6.3}$$

a collection of simplicial subcomplexes. We assume that each L_i is a geodesic ball B_i of some radius r_i. We then define the *size* of a simplicial chain c as

$$r(c) := \min_{i \in \{1,\ldots,k\}} r_i \quad \text{s.t.} \quad c = \partial_{p+1} a + b, \tag{6.4}$$

with $a \in C_{p+1}(\mathrm{K})$ and $b \in C_p(\mathrm{L}_i)$. This ensures that the image of c is trivial in the p^{th} *relative simplicial homology group* $H_{p+1}(\mathrm{K}, \mathrm{L}_i)$. Technically, we define not only the size of a single simplicial chain c, but rather the size of its homology class. Any representative of this class will thus be assigned the same size.

Since we are only dealing with equivalence classes, the cardinality of the equivalence class determines how well our localization scheme can work. An equivalence class that comprises simplicial chains with a very large size, for example, cannot necessarily be made much more localized. This issue is of theoretical interest only—in practice, we experienced large reductions in the size of individual simplicial chains.

Obtaining a localized simplicial chain

Once we have identified the smallest geodesic ball that contains a simplicial chain, we may obtain one representative of the simplicial chain via the *persistence algorithm* described by Algorithm 5 on p. 62. Alternatively, we could follow a suggestion by Chen and Freedman [95], who proposed a greedy algorithm with a runtime of $\mathcal{O}\left(\beta_p n^3 \log^2 n\right)$, where β_p is the p^{th} Betti number and denotes the number of essential homology classes in dimension p, while n is the

number of simplices in the simplicial complex K. However, as we shall see later on in this chapter, our experiments demonstrate that the persistence algorithm works faster in practice. Furthermore, our algorithm may be parallelized and is applicable for weighted Rips graphs, which results in more concise localizations. We will come back to this in a subsequent section.

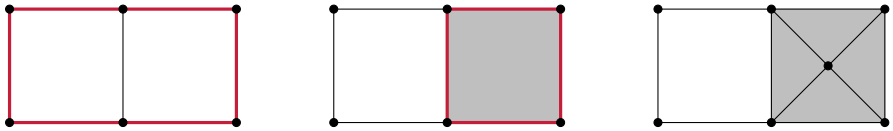
Other localization schemes Our localization scheme is based on geodesic distances, approximated via the Rips graph of a given point cloud. Because this type of algorithm is deterministic, it has an advantage over randomized schemes, such as the one proposed by Zomorodian and Carlsson [410]. Our method is thus suitable for analysing data from recurring experiments. There are other approaches, for example by Dey et al. [127], to obtain localized simplicial chains. These algorithms are restricted to certain kinds of input data, though, such as meshed surfaces, whereas our algorithm is capable of localizing simplicial chains in arbitrary dimensions.

Overview of the localization algorithm

Ideally, we would like our localization scheme to yield simplicial chains whose distances to the boundary of a hole in the data is as small as possible—measured by some set distance, for example. Empirical evidence gained by the analysis of numerous data sets of varying complexities suggests that our deterministic localization scheme satisfies this requirement. To obtain localized simplicial chains in practice, we require the following steps, on which the subsequent sections will expand:

1. Obtain a Rips graph $\mathcal{R}_\epsilon$ using any of the heuristics described in Section 5.4, p. 96 ff. The Rips graph $\mathcal{R}_\epsilon$ is then used to approximate geodesic distances.
2. Expand $\mathcal{R}_\epsilon$ to a Vietoris–Rips complex $\mathcal{V}_\epsilon$ by using Algorithm 1 on p. 49. Extend the approximated geodesic distances from the edges of $\mathcal{R}_\epsilon$, i.e. from the 1-simplices of the Vietoris–Rips complex $\mathcal{V}_\epsilon$ to all simplices of $\mathcal{V}_\epsilon$. This corresponds to describing discrete geodesic balls in $\mathcal{V}_\epsilon$.
3. Find the smallest geodesic ball that contains an *essential simplicial chain* of $\mathcal{V}_\epsilon$, i.e. a homology class of $\mathcal{V}_\epsilon$. Following Definition 6.5, store the size of the simplicial chain. By ensuring that only simplices from within the geodesic ball are used, we localize the simplicial chain.
4. Remove the homology class from $\mathcal{V}_\epsilon$ (a process known as *sealing* because it destroys holes). Repeat these steps until all simplicial chains have been localized.

Figure 6.5 illustrates the individual steps of the localization process by means of a simple topological space. The following sections will describe all steps in more detail.



(a) A simplicial chain (b) A geodesic ball (c) The sealed complex

Figure 6.5: Illustration of the basic localization process. We start with a simplicial chain (a). When calculating the smallest geodesic ball (b), we see that it still supports the simplicial chain, i.e. it still contains an essential homology class. Hence, we use it to obtain a new representative. Finally, we add a 'dummy vertex' and close the geodesic ball (c). This only depicts one potential solution—choosing the left hole would be just as valid.

6.2.1 Approximating & extending geodesic distances

Following Definition 6.1, we need to obtain a geodesic distance function. We observe that each vertex v of the Rips graph $\mathcal{R}_\epsilon$ induces a geodesic distance function by using it as the source vertex in a *single-source shortest paths problem* [113, Chapter 24]. More precisely, we may use Dijkstra's algorithm [113, pp. 658–664], for example, to obtain a graph distance function $\mathrm{dist}_v(\cdot)$. Figure 6.6a illustrates this calculation. By convention, if a vertex w is not reachable from vertex v because they are in different connected components, we define $\mathrm{dist}_v(w) := \infty$.

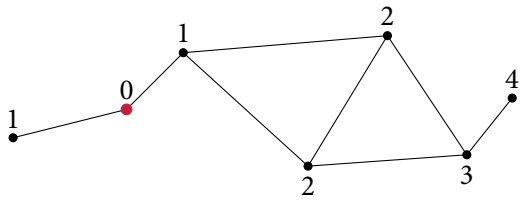
This approximation of geodesic distances was most prominently introduced by Tenenbaum et al. [359] for the Isomap algorithm. Bernstein et al. [41] later proved that a metric induced by graph distances is able to approximate the intrinsic geodesic distances arbitrarily well, provided that the sampling density of the underlying manifold is sufficiently high. In practice, we have of course no way of knowing whether this is the case. This issue is closely-related to the manifold hypothesis again. However, the results of Bernstein et al. justify our proposed localization algorithm and explain why it yields useful simplicial chains. Geodesic approximations are thus often used to analyse complex data sets and may help in revealing relevant structures in data [267, 284].



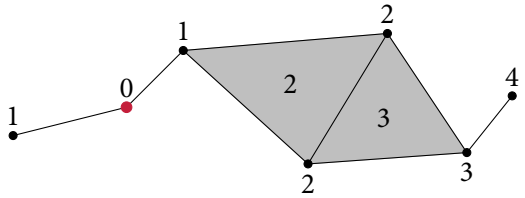
From Lemma 6.2 we know that by extending the geodesic distances to the complete Vietoris–Rips complex $\mathcal{V}_\epsilon$, we obtain a valid filtration. More precisely, we obtain a family of filtrations, indexed by the vertex that is chosen as the source vertex for the geodesic distance calculations. See Figure 6.6b for an illustration.

6.2.2 Finding the smallest geodesic ball

Recalling Definition 6.5, we need to find the smallest geodesic ball that still supports a given simplicial chain. To this end, we calculate geodesic distance functions for every vertex in



(a) Geodesic distances in a Rips graph. Starting from the marked source vertex, we calculate the graph-theoretic distance to all other vertices. In this example, edges have unit weights. However, in contrast to existing algorithms, our localization scheme works for arbitrary weights as well.



(b) Geodesic distances in a Vietoris–Rips complex. The geodesic distances calculated on the 1-skeleton are being extended to higher-dimensional simplices. Here, we only have to deal with 2-simplices.

Figure 6.6: Geodesic distances in graphs and simplicial complexes. After a standard graph distance calculation, the distances are extended in a natural manner. See Definition 4.12 on p. 50 for more details about weight functions and filtrations in simplicial complexes.

Algorithm 13: Finding the smallest geodesic ball

Require: Vietoris–Rips complex $\mathcal{V}_\epsilon$, dimension d

$r \leftarrow \infty$
$\mathrm{B} \leftarrow \varnothing$
for Every vertex $v \in \operatorname{vert} \mathcal{V}_\epsilon$ in random order **do**
    Calculate $\operatorname{dist}_v(\cdot)$ and use it as a filtration for $\mathcal{V}_\epsilon$.
    if $\min \operatorname{dist}_v(\cdot) \geq r$ **then**
        Skip this vertex.
    end if
    Calculate ordinary persistent homology for this filtration.
    $r' \leftarrow$ Creation value of the first essential d-dimensional homology class
    if $r' < r$ **then**
        $r \leftarrow r'$
        $\mathrm{B} \leftarrow \mathrm{B}_v(r)$
    end if
end for
return B

the Vietoris–Rips complex $\mathcal{V}_\epsilon$, resulting in a set of weighted simplicial complexes. For each of these complexes, we calculate persistent homology in the dimension d corresponding to the simplicial chain that we want to localize. We then use the creation value of the first essential homology class in dimension d as the current radius of the geodesic ball—recall that the creation value is the smallest threshold for which the Vietoris–Rips complex has a hole. The minimum of all these creation values yields the centre vertex v and the radius r of the smallest geodesic ball $\mathrm{B}_v(r)$. Algorithm 13 describes this process. In addition to the calculations above, it also extracts the smallest geodesic ball, which we will subsequently use to localize the simplicial chain.

The localization merely requires an additional run of the extended persistence algorithm by Zomorodian and Carlsson [410]. Applying this algorithm to the smallest geodesic ball $\mathrm{B}_v(r)$, which is a simplicial complex by Lemma 6.4, we automatically obtain a simplicial chain that only uses simplices from the smaller complex. Following the notation used in Chapter 4, Section 4.4, p. 55 ff., Algorithm 14 briefly summarizes the calculation.

Algorithm 14: Localizing a simplicial chain

Require: Smallest geodesic ball $\mathrm{B}_v(r)$, dimension d

1: $c \leftarrow \varnothing$
2: Calculate persistent homology for $\mathrm{B}_v(r)$.
3: $\sigma \leftarrow$ Creator simplex of the first essential d-dimensional homology class in $\mathrm{B}_v(r)$
4: $c \leftarrow \mathrm{cascade}[\sigma]$
5: **return** c

Implementation details & performance aspects

When implementing the algorithms presented so far, performance can be improved by omitting information whenever possible. The algorithm for calculating ordinary persistent homology, for example, may abort as soon as the first essential homology class in the desired dimension has been detected. Similarly, there is no need to calculate cascades at this point, making the algorithm more efficient. For further improving performance, we instead employ a *parallelization technique*. Smaller simplicial complexes with less than 10^6 simplices permit the parallel calculation of geodesic distances for multiple vertices. Another speed-up is given by a *branch-and-bound* strategy: When calculating the geodesic distance function dist_v for some vertex v, we may immediately stop and skip the calculation if all updated weights for the corresponding Vietoris–Rips complex are larger than the currently-known minimum radius. Using randomized vertex traversal, our algorithm has a fair chance of finding the minimum radius early, thereby saving needless calculations. Figure 6.7 shows that our simplified algorithm is capable of beating an optimized algorithm by Chen and Freedman [95]. We used the same

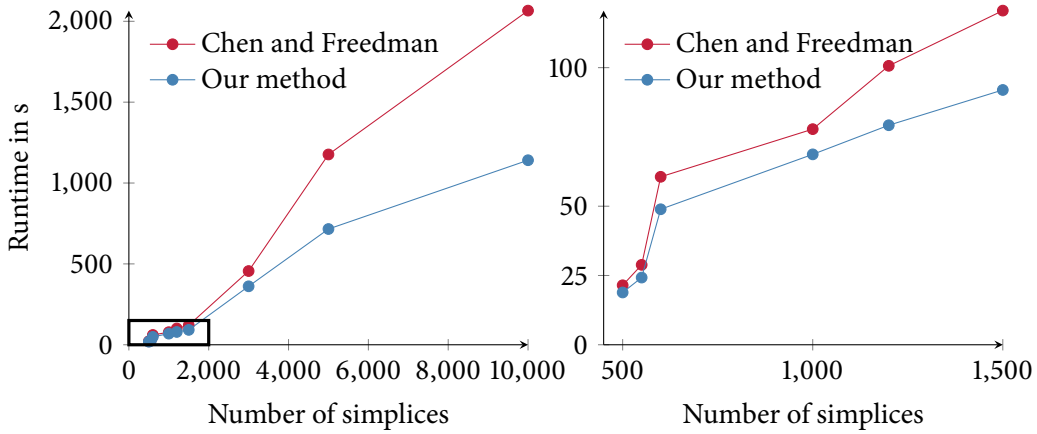


Figure 6.7: Performance results for our localization algorithm. We localized simplicial chains on random samples of a synthetic torus. The improved runtime of our method is especially visible for larger data sets. To focus on the performance for smaller data sets, the plot on the right depicts a magnified version of the region marked in the plot on the left.

optimization techniques in both implementations, but—as outlined above—our algorithm benefits from a simpler localization strategy. Moreover, the other algorithm only works for simplicial complexes with unit weights, whereas our algorithm has no such constraints.

6.2.3 Removing a homology class from $\mathcal{V}_\epsilon$

After localizing a simplicial chain in dimension d, we need to augment the Vietoris–Rips complex $\mathcal{V}_\epsilon$ such that the simplicial chain stops being an essential homology class. Else, we would not be able to localize more than one simplicial chain per dimension. To this end, we add a new 'dummy vertex' v to $\mathcal{V}_\epsilon$. For each simplex σ in the simplicial chain, we add a $(d+1)$-simplex with v as an additional vertex to $\mathcal{V}_\epsilon$. We also add all faces of these new simplices. This ensures that the hole that is bounded by the simplicial chain is being closed and does not appear in subsequent calculations. See Algorithm 15 for a description.

6.3 The simplicial chain graph

So far, we have seen how to localize simplicial chains in a data set. These localized chains serve to describe both the geometrical and the topological structure of data. We thus want to visualize them to obtain their structural information. This visualization is challenging because even a low-dimensional simplicial chain cannot be visualized for a high-dimensional data set, as every vertex of its simplices still corresponds to a high-dimensional data point.

Algorithm 15: Sealing a simplicial complex

Require: Vietoris–Rips complex $\mathcal{V}_\epsilon$, simplicial chain c

Choose a 'dummy vertex' v.
$\mathcal{V}_\epsilon \leftarrow \mathcal{V}_\epsilon \cup \{v\}$
for Every simplex $\sigma \in c$ **do**
    $\sigma' \leftarrow \sigma \cup \{v\}$
    $\mathcal{V}_\epsilon \leftarrow \mathcal{V}_\epsilon \cup \{\sigma'\}$
    for Every face $\tau \in \sigma'$ **do**
        $\mathcal{V}_\epsilon \leftarrow \mathcal{V}_\epsilon \cup \{\tau\}$
    end for
end for
return $\mathcal{V}_\epsilon$

Simple connectivity visualization

Instead of attempting to use positional information of the data points, we shall rather focus on their *relations*, which we depict as a graph. In order to make the graph layout reproducible, we need to use a deterministic layout algorithm such as `neato` [174] with suitable initialization parameters. A simple approach towards the layout would assign simplices to graph nodes and connect them whenever they are a part of the same simplicial chain. Figure 6.8 depicts a typical output. This sort of graph serves to give a rough overview of all available simplicial chains but ultimately, it does not convey much information and has severe shortcomings:

- The relation between a simplex in the graph and the corresponding subset of input data is not apparent. In particular, small distances in the graph do not correspond to small distances within the original data set.
- Different simplicial chains cannot be discerned from each other if they have a common simplex. Begin and end of a chain are not indicated.
- Since nodes can be part of multiple simplicial chains, information about a complete chain cannot be encoded via their visual attributes.

Refining the graph

We thus need to integrate information about both the chains and the input in the graph in order to obtain a structural description of a multivariate point cloud via its simplicial chains. To this end, we first *decompose* each simplicial chain into the data points it contains. This is done by inserting each vertex of each simplex of the chain into a set. We then associate this set with the corresponding high-dimensional input coordinates and call this transformation the *coordinate decomposition* of a simplicial chain. See Algorithm 16 for more details. Having

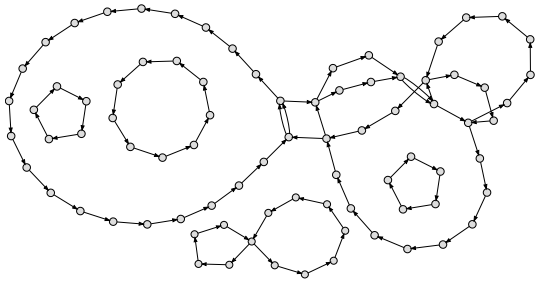


Figure 6.8: Graph visualization of simplicial connectivity relations. This sort of visualization serves to give only a very rough overview of the data, conveying almost no additional information.

applied this decomposition to all simplicial chains, we create the *simplicial chain graph* as a graph with two types of nodes:

1. *Chain nodes* that correspond to a simplicial chain in the data set.

2. *Data nodes* that correspond to a data point in the input data set. The simplicial chain graph only contains a data node for points that occur in at least one simplicial chain. Hence, only points that contribute to the topological features will be shown.

We add an edge between a chain node and a data node whenever the corresponding simplicial chain contains the corresponding data point. The degree of a data node thus represents the number of simplicial chains it is a part of. Note that there are no edges between chain nodes—only the data nodes are used to show relations in the data set.

Algorithm 16: Coordinate decomposition of a simplicial chain

Require: Simplicial chain c
1: vert $c \leftarrow \varnothing$
2: **for** All simplices $\sigma \in c$ **do**
3: **for** All 1-faces $\{v\} \subseteq \sigma$ **do**
4: vert $c \leftarrow$ vert $c \cup \{v\}$
5: **end for**
6: **end for**
7: $\mathrm{C} \leftarrow \varnothing$
8: **for** Every index $i \in$ vert c **do**
9: Look up the corresponding point $p_i \in \mathbb{R}^n$ in the input data.
10: $\mathrm{C} \leftarrow \mathrm{C} \cup \{p_i\}$
11: **end for**
12: **return** C

Adding distance information

We introduce an additional geometrical element to the graph structure of the simplicial chain graph. This first requires us to define a representative for the coordinate decomposition of each simplicial chain.

Definition 6.6 (Medoid). Given a finite set $\{x_1, \ldots, x_n\}$ of points in $\mathbb{R}^d$ and a distance function $\mathrm{dist}(\cdot,\cdot)$, the *medoid* is the element m that minimizes the objective function

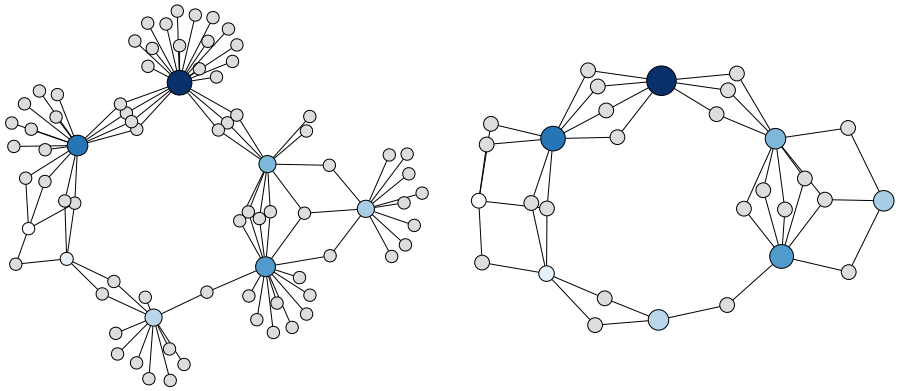
$$\frac{1}{n}\sum_{i=1}^{n}\mathrm{dist}(x_i, m), \tag{6.5}$$

i.e. the element whose average dissimilarity to all points is minimal. Since the set is assumed to be finite, the medoid always exists. In contrast to the geometrical mean, for example, the medoid is always a member of the input data.

After decomposing each simplicial chain via Algorithm 16, we calculate the medoid of its coordinates. The set of medoids then permits us to define distances between chain nodes. Given two chain nodes c_1 and c_2, their ideal distance is $\mathrm{dist}(m_1, m_2)$, where m_1 and m_2 are the medoids of the two simplicial chains. The graph layout algorithm aims to place chain nodes such that these ideal distances are preserved as much as possible in the rendered graph. The distances help maintain a sense of geometrical similarity between different substructures in the graph. In particular, simplicial chains that are close in the input space will be placed in close proximity.

Colour-coding

Since the simplicial chain graph consists of two different node types, we can use colours to encode further topological or geometrical attributes. By default, we follow Definition 6.5 and colour all chain nodes by their size using a continuous colour map in which darker colours indicate larger sizes. Figure 6.9, left, depicts a typical example. The graph only has a single connected component but numerous chain nodes, i.e. coloured nodes. This indicates that the holes in the data are not too far removed from each other. At the same time, the large empty area that is surrounded by the graph shows that no data exists within this region. Else, the region would contain at least some chain nodes with a smaller size. The large radius and the dark colour of one chain node reveals the existence of a simplicial chain with a large size. The size of a simplicial chain is correlated to the amount of space a topological feature encompasses in the input data. A simplicial chain with a large size thus corresponds to a structure that is spread out over a large part of the data, while smaller sizes indicate structures that are more local.



Uncompressed simplicial chain graph Compressed simplicial chain graph

Figure 6.9: Uncompressed and compressed simplicial chain graphs. Both chain graphs indicate the presence of at least one hole with a large size, but the compressed graph is less cluttered.

The colour-coding may of course be changed to represent another attribute, such as the different notions of the *volume* of a simplicial chain. [94, 95], provided the selected attribute is useful for comparing different data sets.

Removing clutter

Although the colour-coded simplicial chain graph serves to highlight structures in a point cloud, it gets increasingly cluttered the larger the input data sets become. We thus need to compress the graph. To this end, we remove all data nodes (and their associated edges) with a degree of 1, which means we only keep a data node if it is a part of multiple simplicial chains.

We encode the cardinality of the simplicial chain by linearly scaling each chain node accordingly. The compression effectively reduces the visual complexity of the graph. Figure 6.9, right, shows an example of the *compressed simplicial chain graph*.

6.3.1 Properties

The simplicial chain graph describes the most important (in terms of the persistence) parts of the topological structure of a data set. It has the following key properties:

1. *Homogeneity measure*. Both the number of connected components and the distances between different chain nodes in the graph are directly correlated with the *homogeneity* of a data set. A large amount of simplicial chains that share no data points indicates that the data set contains multiple regions and is thus not a homogeneous entity. A simple example is a data set with one class of measurements lying on a hypertorus, and the

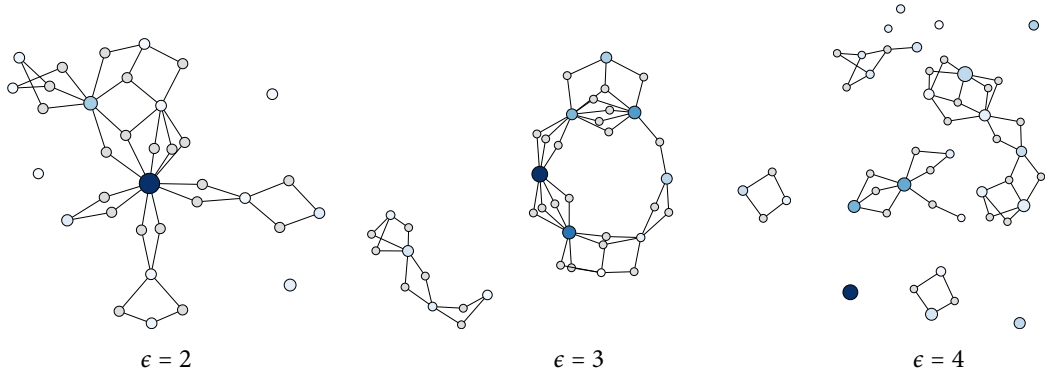


Figure 6.10: Simplicial chain graph stability. Large variations in the ϵ parameter will result in different structures being depicted by the simplicial chain graph.

other one lying on a hypersphere. Such a configuration will show up as two connected components in the simplicial chain graph.

2. *Size distributions*. The *size distributions* of the chain nodes encode the sizes of the individual geometrical–topological substructures in the data set. This information is useful when comparing multiple data sets. If two data sets are created from the same underlying phenomenon, for example, their size distributions should be equal. A large difference may indicate an error in the sampling, meaning that there are insufficient measurements in one data set.

3. *Empty region measurements*. The sizes of the simplicial chains indicate the size of inhomogeneous regions in the data. A single simplicial chain with an extremely large size bounds a large empty region in the data. In case of experimental measurement data, this might indicate missing values, for example.

In Section 6.4, we shall see how these properties may be put to use for the analysis of several data sets. In particular, we will use the simplicial chain graph to guide EDA.

6.3.2 Stability & extensions

The simplicial chain graph is not limited to the visualization of simplicial chains of a single dimension. The data sets from Section 6.4, for example, contain both 1-dimensional and 2-dimensional simplicial chains. For the calculation of the simplicial chain graph, we only require a list of dimensions for homology localization and a value for ϵ. This parameter is used to control the scale for calculating persistent homology. Depending on ϵ, different scales may be emphasized in the data set. Varying ϵ might thus cause the simplicial chain graph to change. Figure 6.10 shows large changes in the structure of the graph that are caused by large

variations of ϵ. For smaller perturbations, however, the *stability theorem* of Cohen-Steiner et al. [104] implies that the simplicial chains remain stable. To select a suitable value for ϵ, we can employ any of the heuristics from Chapter 5, Section 5.4, p. 96 ff.

✦

Note that the simplicial chain graph only shows the homology classes of $\mathcal{V}_\epsilon$, i.e. topological features with infinite persistence. The reason for this restriction is that the localization of arbitrary simplicial chains is still an open problem. The simplicial chain graph thus only focuses on the most prominent topological features of a data set for now. Although it would be possible to accommodate simplicial chains of finite persistence in the simplicial chain graph, there is no guarantee that such chains describe salient structures in the data.

6.4 Analysis of several data sets

We shall now see how to use the simplicial chain graph for the analysis of several high-dimensional data sets from different application domains. For setting ϵ, we employ the *nearest neighbour heuristic* described in Chapter 5, Section 5.4, p. 96 ff. As the selected data sets only exhibit few higher-dimensional topological features, we shall restrict the analysis to 1-dimensional and 2-dimensional simplicial chains.

6.4.1 Voting data

One of the aims of political science is to analyse voting patterns in the general public and in democratic representatives over time. A typical feature of parliaments is the creation of *voting blocs*, i.e. smaller subsets of representatives whose voting behaviour is very similar. As the political system of the United States of America is essentially a two-party system, a large body of knowledge deals with the effects of different variables on the voting behaviour of representatives. Early studies [64] concentrated on finding out whether the constituency, i.e. the electoral district of a representative, affected the voting behaviour more than the party affiliation. Tabulating the required data for this sort of political analysis was rather cumbersome until public interfaces were introduced that made obtaining these data more easy. Jakulin et al. [212], for example, use cluster analysis to uncover voting blocks in the United State Senate. In an earlier work, Porter et al. [297] apply network analysis to detect committees in the United States House of Representatives. They find that the party affiliation is the dominating structure of the data that tends to cloud other features. Consequently, we want to apply topological analysis—with its inherent multi-scale feature detection—to

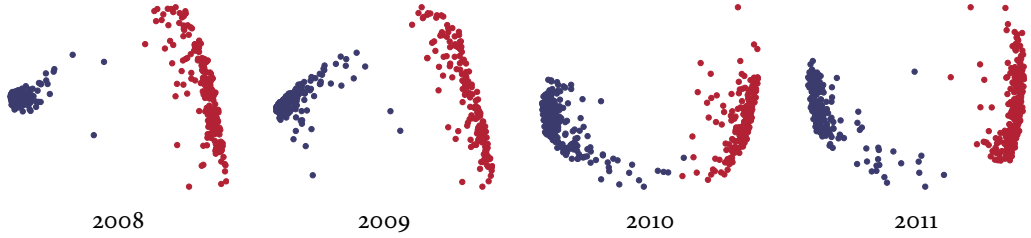


Figure 6.11: PCA projections of congressional votes. Each point corresponds to some representative. The party affiliations are dominant—there is a clear divide between Democrats ● and Republicans ●. We can see that the shape of the data varies over time.

uncover more structural features of the data. Our analysis goes beyond these and related approaches [254] because we detect and visualize topological features in the raw data.

Pre-processing

We obtained the results of different votes cast by the United States Congress in a period from 1990–2011. To convert the roll call votes to a point cloud, we modelled the different outcomes—'Yea', 'Nay', or abstention—as either +1, −1, or 0, resulting in a long vector for each representative. Depending on the number of votes in each session of Congress, we obtain data sets with about 420 instances (each corresponding to a representative of some state) and 600–900 dimensions (each corresponding to the result of a vote on a certain topic). Porter et al. [297] used data in such a format to obtain a voting matrix for which they calculated a *singular value decomposition* (SVD). The SVD then permitted the classification voting positions of the representatives.

In a similar vein, Figure 6.11 depicts PCA projections for different time periods. The party affiliation may be easily discerned from each projection, but obtaining more information than this dominant signal requires different tools. Our topological analysis focuses more on the overall shape of the data. We want to find those substructures that define the data in a sense, i.e. substructures whose removal would cause the data to change its shape. For the subsequent analysis, we shall use the *Hamming distance*, as it is very suitable for describing distances in this space [254, 381].

Definition 6.7 (Hamming distance). Let $x = (x_1, \ldots, x_k)$ and $y = (y_1, \ldots, y_k)$ be two vectors describing the voting behaviour of two representatives, where $x_i, y_i \in \{-1, 0, +1\}$. The *Hamming distance* between x and y is defined as

$$\operatorname{dist}_{\text{Hamming}}(x, y) := \operatorname{card}\left\{ i \in \{1, \ldots, k\} \mid x_i \neq y_i \right\}, \tag{6.6}$$

i.e. the number of disagreeing votes. The Hamming distance considers all differences to have an equal weight—an agreement and an abstention thus have the same distance as an agreement and a disagreement.

Topological features in the voting data

When analysing the votes using the Hamming distance, we should first think about what topological features we can expect in the data. If all representatives of a given party were to vote exactly identically, there would be no topological activity in the data because it would contain only two points. Every topological feature is hence created by disagreeing votes. Consequently, a large amount of topological features in one party indicates that the party is divided about numerous issues.

Our analysis uncovers a simple structure for the space of representatives. We do not find any non-trivial topological activity with a dimension $d \geq 3$, which suggests that the data have a low intrinsic dimensionality. This confirms established results by Poole and Rosenthal [296]. In the following, we will exemplarily take a look at voting data from 2008 and 2009 as these data sets contain the largest number of votes of the whole period.

Figure 6.12 uses *sorted heat maps* to show the general structure of the data. Red indicates opposing votes, green indicates approval, and white indicates abstention. The votes of each individual Representative are shown in the rows of the heat map. The block structures of the same colour show that there is a clear distinction between the two parties. Some issues or bills are approved of by both parties equally, though, apart from some dissenters.

Definitions

Prior to describing more detailed results for two example voting periods, we first need to define some terms. We do not presume to be experts in political analysis but instead want to show the utility of this novel form of analysis.

Definition 6.8 (Party line). The concept of *party line* usually refers to the political aims that are pursued by a certain party. We take a pragmatic approach and define the party line to be the vote of the majority of all party members. For example, if the majority of representatives opposes a certain issue, we consider this opposition to be the official party line.

Definition 6.9 (Maverick score). We refer to the number of times a representative votes in direct opposition to the party line as the *maverick score*. We use this term somewhat tongue-in-cheek; political science considers a *maverick politician* to be someone who 'votes their conscience'. Our analysis cannot account for any of these factors.

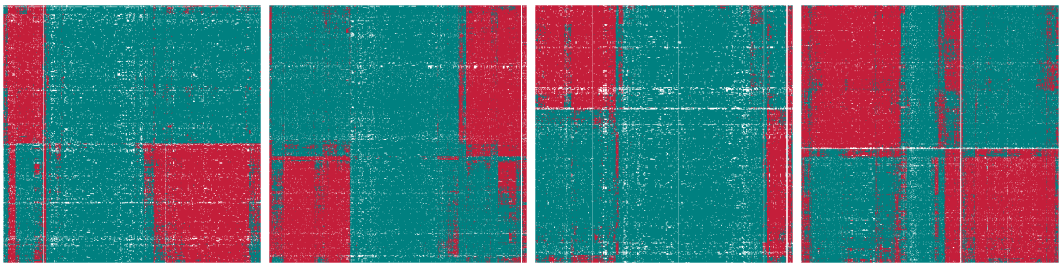


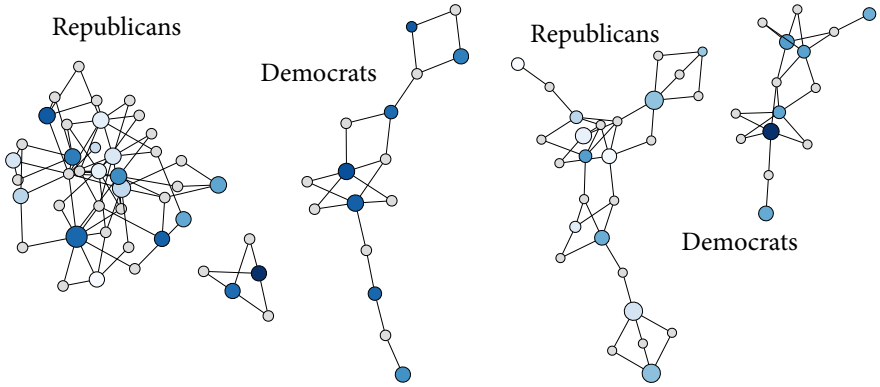
Figure 6.12: Sorted heat maps of the roll call results for 2008–2011. Party 'blocks' can easily be identified. The amount of agreement on both sides for certain issues varies between the years. Red indicates opposing votes, green indicates approving votes, and white indicates abstention.

With these definitions, we may now exemplarily talk about two different voting periods in Congress. The first period—2008—was the heyday of the economic crisis, while the second period—2009—was the first year of Barack Obama's presidency. We used $\epsilon = 80$ and the *Hamming distance* for the subsequent analysis.

Analysis of 2008 votes

Figure 6.13a shows the simplicial chain graph of the 2008 data. We can see that it shows two larger connected components, one for the Democratic Party, the other one for the Republican Party. This is an expression of the inhomogeneity that is inherent to the data and indicates the different voting behaviour of Democrats and Republicans. The simplicial chains in each connected component correspond to a set of representatives with similar voting behaviour. One of the larger simplicial chains, for example, comprises the Republican Representatives Buchanan, Dent, Kuhl, McCotter, and Platts. All of these representatives feature comparatively high maverick scores—Dent has one of 89 and Platts has one of 99, for example. For the Democrats, similar observations apply, although their maverick scores are considerably lower for these data. One of the simplicial chains features representatives Kildee, Matsui, Moore, Pomeroy, Spratt, Stupak, and Thompson. None of these has a maverick score higher than 34. These simplicial chains are thus a description of the boundary of the full party structure. The smaller connected components, by contrast, describe smaller subsets of representatives (only about 2–3) whose voting behaviour is similar but does not coincide with larger groups of mavericks of their respective parties.

It is interesting to note that Democratic Representatives appear to have a large unity with respect to their voting behaviour than the Republican Representatives. This is shown by the smaller amount of topological features in the connected component of the Democrats. Nodes in this connected component also tend to have smaller sizes than the nodes in the



(a) 2008 (b) 2009

Figure 6.13: Simplicial chain graphs of the roll call results for 2008 and 2009. The size differences between the structures are partially explained by the different number of votes.

remaining components. This means that the sets of Democratic Representatives with similar voting behaviour have a smaller cardinality than those of the Republican Representatives. In essence, Democrat votes appear to be less fragmented. Moreover, there is less overlap in voting behaviour, resulting in a long strand of chain nodes. The Republican Representatives form a tighter cluster with more edges. This illustrates that there is a larger overlap between dissenting votes, i.e. votes that are different from the party line.

We see that the changes in voting behaviour give rise to well-defined topological features, whose analysis uncovers relations that go beyond the mere party affiliations. Nonetheless, the simplicial chain graph also expresses the 'party divide' through its connected components.

Analysis of 2009 votes

The data for 2009, shown in Figure 6.13b, exhibits a similar behaviour. Since there are slightly more votes than in the 2008 data set, the decomposition into exactly two large structures is seen more clearly. The smaller connected component corresponds to the Democratic Party, which held the majority of seats in 2009. A better detection of the boundary of the party, resulting from more representatives disagreeing with the official party line, is thus to be expected.

Analysing some simplicial chains, we find that the maverick scores of Democratic Representatives are significantly higher than in 2008—one of the largest chains contains representatives Kennedy and Lewis with maverick scores of 209 and 210, respectively. Nonetheless, the small amount of topological features indicates that the Democrats exhibit a high degree of conformity with respect to their party line, just as they did in 2008.

The Republican Representatives appearing in the simplicial chains consistently feature lower maverick scores. With 110 and 114, representatives Bachus and Dreier have the highest scores for their party. In contrast to the 2008 votes, the Republican Representatives also appear to have a higher degree of voting unity here, as indicated by their connected component, which starts to become more elongated. As a consequence, the homogeneity of this voting period is different. We furthermore note that the size distribution of simplicial chains is different than in the previous period. The Republican Representatives hence exhibit a lower amount of dissenting votes.

Summary

We conclude that the simplicial chain graph shows that the voting data sets have a clear topological shape. Said shape is determined by all representatives whose votes disagree with their respective party lines. Each connected component shows the voting behaviour of a subset of the representatives, while each simplicial chain describes a subset of Representatives with *similar* voting behaviour. These subsets of representatives shed light on the political climate and could conceivably be used to detect structures such as *voting alliances*. Furthermore, these substructures make up the boundary and the shape of the votes of each party. By calculating simplicial chain graphs for longer periods, it would be possible to show the evolution of voting behaviour of the United States Congress.

6.4.2 Tropical Atmosphere Ocean array data

The El Niño phenomenon refers to a pronounced climate pattern that is defined by prolonged anomalies of sea surface temperatures in the Pacific Ocean [366]. El Niño typically occurs at irregular intervals from 3–7 years and may last from 9 months to 2 years. The mechanisms that cause this phenomenon are still a topic of research. Better insights into the formation and the properties accompanying El Niño are necessary to prevent damages by flash floods and dry periods, for example. Other influences, such as man-made climate change, make predicting the effects of El Niño even harder [110].

We use data from the Tropical Atmosphere Ocean (TAO) array [194]. This array consists of approximately 70 buoy moorings in the Tropical Pacific Ocean; see Figure 6.14 for an overview of the area. Each buoy records five-dimensional measurements, consisting of zonal wind velocity, meridional wind velocity, humidity, air temperature, and sea surface temperature. We obtained a set of measurements for a period from 1980–1998. El Niño occurred several times within this period, namely 1982–1983, 1986, 1991–1992, 1993, 1994–1995 and 1997–1998. Over a period of 18 years, the data comprises about 180,000 data points. Missing values make analysing these data very challenging. Because of technical errors and different buoy

Figure 6.14: Region of buoy moorings in the Pacific Ocean. Buoys are moored north of the coast of Papua New Guinea as well as south of the coast of Nicaragua. The effects of the El Niño phenomenon are most noticeable in this region.

configurations, not all attribute measurements are available for the whole recording period. Figure 6.15 and Figure 6.18 depict some typical examples of simplicial chain graphs for different time periods. The remaining time periods exhibit similar patterns. We selected a time period from 1993–1997 because they alternate between El Niño events and regular measurements. Moreover, 1997 saw the largest El Niño event on record so far.

Topological features in the TAO data

Prior to calculating and analysis the TAO data, we should again briefly point out what types of topological features the data may exhibit. Similar to the analysis in Section 6.4.1, the point cloud would not exhibit any topological features if all measurements were equal. Differences in measurements thus give rise to high-dimensional holes, whose sizes correspond to the magnitude of the difference. *Missing measurements* also yield holes because the creators of the data set encode them by extremely low values, resulting in a large distance between data points. Similarly, large-scale changes of attribute values—such as the changes brought on by El Niño—create high-dimensional holes. The different ways by which holes are created in the TAO data hence justify the utility of our topological approach.

Analysis of 1993 data

We initially focus on the simplicial chain graph for the 1993 period, as shown in Figure 6.15, left. It shows a single large connected component (a), a small connected component (b), and some isolated nodes (c). We have adjusted the distances between the components for layout reasons. The original graph exhibits larger distances between the individual connected component, which indicates that the corresponding simplicial chains describe different characteristics of the corresponding buoy measurements.

We first take a look at the largest connected component (a). It contains two large empty regions, indicating large differences or variations between individual measurements. Fig-

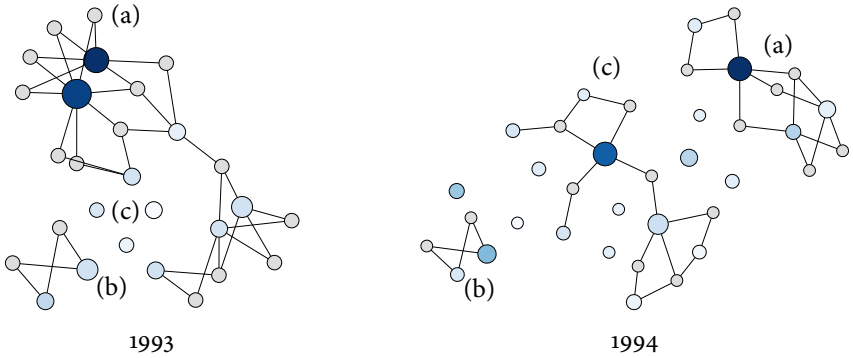


Figure 6.15: Simplicial chain graphs of the TAO data (1993–1994). The large connected component in the 1993 chain graph gets split in the 1994 graph.

ure 6.16a shows a PCP of some of its simplicial chains; we observe that they consist of points with medium–low zonal and meridional wind velocities and high values for humidity, air temperature, and sea surface temperature. At least two of the variables, namely the wind velocities, exhibit a high variance. By contrast, Figure 6.16b shows that the connected component (b) contains points with slightly larger zonal wind velocities than component (a). Apart from that, the profiles are rather similar. The variance of all variables is smaller, leading to a smaller size of the simplicial chain when measured according to Definition 6.5. Finally, the isolated nodes (c) turn out to correspond to measurements with missing humidity values and/or missing temperatures—as depicted in Figure 6.16c. The colour of the nodes in the chain graph indicates that these are topological features with a small scale.

This example illustrates that the input space contains several topologically-distinct regions, which are captured in a meaningful manner by the simplicial chain graphs. These regions cannot be captured as easily by distance-based methods because the measurements do not form pronounced clusters.

Analysis of 1994 data

Figure 6.15, right, depicts the simplicial chain graph for the 1994 period. It exhibits a different set of patterns. By interacting with the graph, we first attempt to match its connected components to the components we have encountered in the 1993 data. It turns out that the larger connected component from the 1993 chain graph is split up into two connected components. Both components (a) and (c) are similar to each other and similar to the connected component (a) in the 1993 chain graph.

PCPs show that they describe similar phenomena: Figure 6.17a shows that component (a) is characterized by low zonal wind velocities, medium meridional wind velocities, and extremely low humidity values, while the temperatures are extremely high. These large variations result in

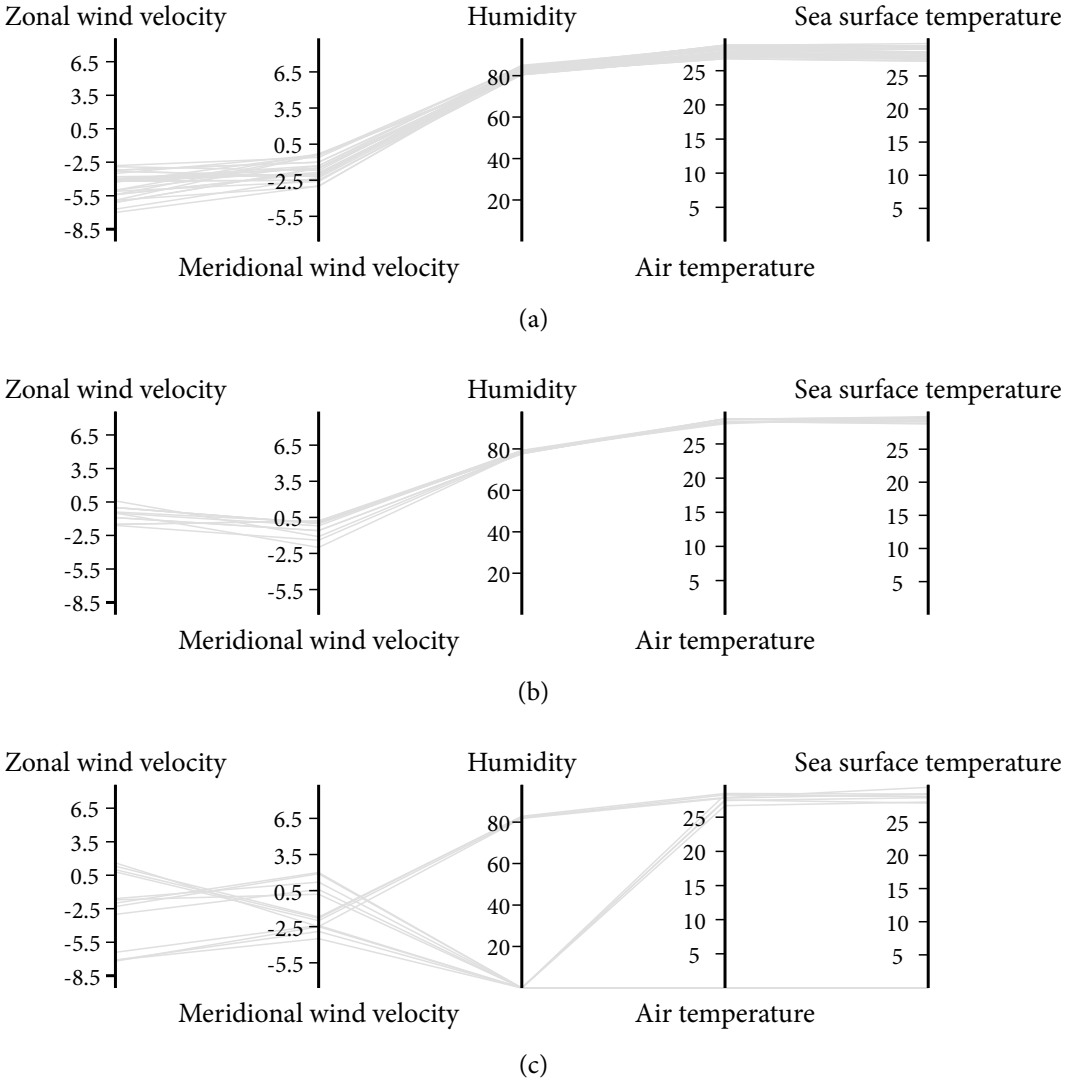


Figure 6.16: Visualizations of simplicial chains for 1993. The chains bound different regions in the attribute space. Each region is characterized by a unique profile of values.

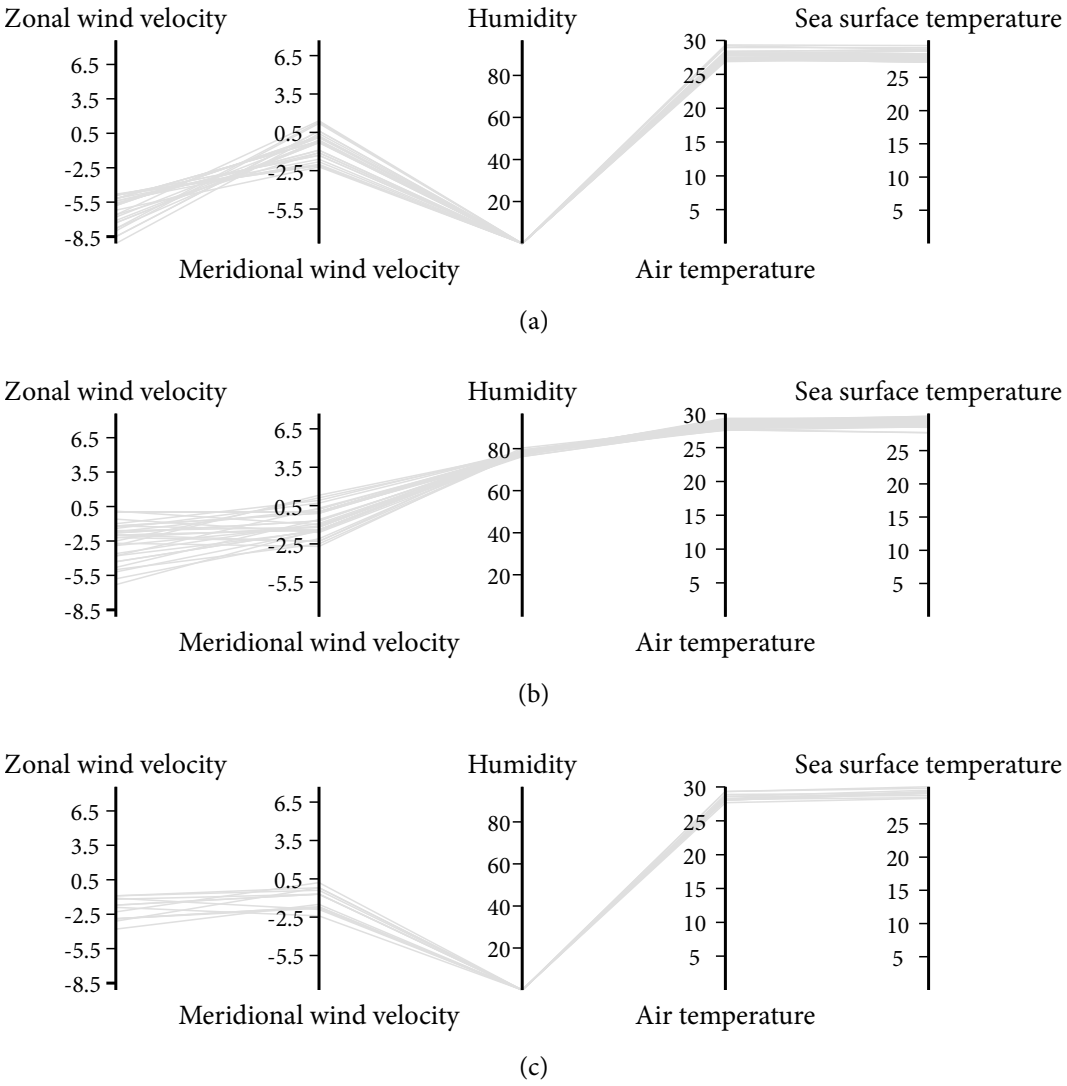


Figure 6.17: Visualizations of simplicial chains for 1994. The appearance of El Niño results in marked topological changes—temperatures tend to be higher, for instance.

a simplicial chain with a large size. The radius of the corresponding node in the simplicial chain graph indicates that the simplicial chain comprises more measurements than the remaining chains. A further analysis of the anomalously low humidity values shows that they are missing for the specified time period. Figure 6.17b shows that component (b) has virtually the same profile except for the humidity values, which are almost extremal. Finally, component (c), as depicted in Figure 6.17c, corresponds to anomalous points with higher zonal wind velocities than the larger components. This is similar to the behaviour in the 1993 data.

We also observe that the temperature attributes of the points in the 1994 simplicial chain graph are slightly higher than the temperatures for the 1993 data. This effect cannot be seen in the PCPs due to the scaling. The split into multiple connected components indicates that the shape of the data undergoes a severe change. This change is caused partially by the missing values for the time period, but also by the extremal temperature values, which are a characteristic feature of El Niño.

Analysis of other time periods

The same anomalous changes in the measurements are reflected in the simplicial chain graphs for 1996 and 1997, shown in Figure 6.18. Especially the last data set has many missing values in all attributes, resulting in a significantly fragmented attribute space. At least for the 1996 data, the simplicial chains still exhibit similar characteristics as in the previous data sets. The 1997 simplicial chain graph, on the other hand, looks markedly different from the other simplicial chain graphs. This is caused by an even larger number of missing measurements, which makes the extraction of information very difficult. Even though our visualization technique is stable—see Section 6.3.2—we cannot extract the topology with the same fidelity as for the other time periods.

Seen in context with the other data sets, the simplicial chain graph suggests that the 1997 data contains more anomalies than the other data sets because it exhibits a different topological structure. The amount of anomalies in multivariate data cannot be detected as easily by standard visualization methods. The simplicial chain graph thus yields a quick overview of qualitative changes in the data at the expense of a higher abstraction level.

6.5 Discussion

In this chapter, we introduced a new algorithm that endows persistent homology calculations with more geometrical information. In contrast to earlier algorithms, our method makes use of the weights in a simplicial complex, is easy to implement, and may be parallelized. We also introduced *simplicial chain graphs*, a new visualization of the connectivity structure of high-

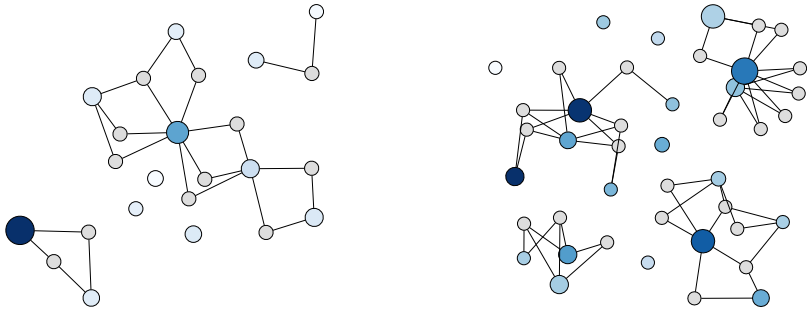


Figure 6.18: Simplicial chain graphs of the TAO data (1996–1997). For 1997, many measurements are missing, making the extraction of geometrical or topological information difficult. This shows up as an increased amount of connected components and chain nodes with a large size, as indicated by the darker colour.

dimensional point clouds. Although the simplicial chain graph yields a rather abstract view of data, it helps depict anomalous structures and quantify similarities. Combined with standard interaction techniques, such as brushing+linking [63, 130], as well as standard multivariate visualizations, such as PCPs [210] or SPLOMs [86], the simplicial chain graph is a useful addition to a multivariate data analysis toolbox.

Limitations

The large runtime makes the localization of all simplicial chains still prohibitive for larger data sets. Our method would hence profit from any parallelization strategies for persistent homology, such as the approach by Lewis and Morozov [243], or approximative strategies, such as the ones by Sheehy [334], Buchet et al. [61], or Cavanna et al. [82], who give a geometrical perspective on vertex removal. Another possibility would be to use regular simplicial chains, i.e. the cascades created by the persistent homology algorithm. Recent work by Bendich and Bubenik [34], however, shows that these cascades tend to be unstable. Hence, our localization strategies are justified.

Extensions

The simplicial chain graph may potentially be used for the analysis of time-varying data with a finer resolution of time-steps. As previous research in the context of time-varying graphs shows [230], it is still somewhat of a challenge to obtain stable representations here. In particular, focus preservation requires special provisions here [164]. Likewise, there are still unresolved aspects concerning the description of topological changes of a function over time. To this end, Edelsbrunner et al. [143] presented an algorithm that maintains

Reeb graphs. By means of a smart data structure, it is hence possible to extract the Reeb graph at a given time-step. The visualization of individual Reeb graphs then helps explain certain phenomena in the data. In the setting of persistent homology, Cohen-Steiner et al. [106] developed *persistence vineyards*, a way of expressing changes over time in persistence diagrams. The visualization of individual *persistence vines* turns out to be difficult, but vines appear to provide salient information about the behaviour of highly-complex systems. This was further examined by Morozov [272], but vines have only more recently seen applications in time-varying systems [276]. A contrasting approach by de Silva et al. [339] analyses time-delay embeddings of signals. Building on this, Perea and Harer [292] developed a theoretical persistence-based framework to quantify periodicity of these signals. Related publications [126, 291] showed that this is a useful description for many applications, such as the analysis of gene expression data. The author considers a different approach to be equally fruitful. Future research should first focus on useful *summary statistics* of persistence diagrams that are easy to calculate, such as the p-norm. By graphing and analysing—with the help of standard tools for time-series analysis—these summary statistics over time, patterns in the topological structure of data already become apparent.

Further research also should focus on deriving more measures of the geometry of simplicial chains. Previous work [94, 95] only mentions simple measures, such as the sum of weights in a chain. More elaborate measures that take the 'shape' of the simplicial chain better into account are needed. It would be interesting to see whether certain hulls, such as the *convex hull*, the *concave hull*, or *alpha shapes* [147, 152] are salient shape descriptors even in high dimensions.

Another interesting venue of research involves the study of useful embellishments to the simplicial chain graph. The chain nodes in the graph could be endowed with glyphs, for example, that represent the connectivity of the corresponding simplicial chain. Since simplicial chains are essentially nothing else than graphs, methods from spectral graph theory [58, 102] may turn out to be useful here. The author considers a visualization of the graph Laplacian of a simplicial chain to be a salient descriptor of the connectivity with sufficient explanatory power. As spectral methods are nowadays also often used in graph drawing [25], they show some potential as a visualization tool.

Part II

Visualizing quantitative topological information

The second part of this thesis focuses on visualization techniques for *quantitative topological information* of data. We will use persistent homology as a way of extracting features from multivariate data. These features permit us to quantify precisely to which extent certain properties are present in data. The methods contained in this part hence work best in scenarios that require assessing data either explicitly or implicitly, such as when users need to select suitable clustering algorithms.

7 Evaluating embeddings

Dimensionality reduction methods are commonly used to make sense of multivariate data sets. In the last few decades, the field has moved from using simple linear methods such as *principal component analysis* (PCA) [219] towards an ever-increasing amount of specialized dimensionality reduction methods of varying complexities. Most of these methods have the underlying assumption that data are sampled from an unknown manifold $\mathcal{M}$. The basic idea is to use some of the intrinsic properties of $\mathcal{M}$, such as its convexity, in order to obtain an embedding into a lower-dimensional space. Two types of embeddings are very common. The first reduces data from a very high-dimensional space in $\mathbb{R}^d$ to a lower-dimensional space in $\mathbb{R}^k$ with $k \ll d$. This reduction is again a manifestation of the *manifold hypothesis* we encountered in Chapter 2, Section 2.6, p. 18. In this case, working in $\mathbb{R}^k$ is tantamount to working with a compressed version of the data. The second type of embedding is motivated by the principles of *exploratory data analysis* (EDA), as pioneered by Tukey [369, 371]. Here, the unknown manifold $\mathcal{M}$ is to be embedded in $\mathbb{R}^2$ for visualization purposes. Analysts then hope to gain knowledge about the internal structure of the data by looking at these low-dimensional embeddings. Of particular interest are regions of varying density in embeddings. Their occurrence may indicate clusters, for instance.

Regardless of the goals of the visualization, choosing a suitable embedding for processing one's data remains a challenging task. The problem is highly ill-posed and reminiscent of a 'chicken-and-egg' dilemma: While users need to know the internal properties of data, such knowledge is often only gained *after* applying a certain dimensionality reduction method. The fidelity of such information is thus often questionable.



This chapter shows how we may evaluate and assess different embeddings using persistent homology. We will treat this endeavour in two parts. First, after defining a number of common quality measures for dimensionality reduction algorithms, we will use persistent homology to measure their agreement with respect to the properties of a single embedding. This method, which is presented in Section 7.3, will permit us to find out which properties of high-dimensional data—such as its local neighbourhoods—have been retained in an embedding. Second, we will show how we can employ a novel variant of persistent homology

in order to evaluate multiple embeddings. To this end, Section 7.6 introduces defines a new measure for assessing the amount of distortions present in an embedding. Based on this assessment, we will show the user both global and local errors in an embedding. We will proceed by using various sources of data and multiple embeddings. Our goal is to find the most suitable method for visualizing the data in $\mathbb{R}^2$. Here, 'most suitable' refers to the method that exhibits the least amount of distortions. The contents of this chapter are based on two publications [310, 315], which contain preliminary results about the agreement analysis and the generic evaluation of embeddings.

7.1 Dimensionality reduction methods

In the following, we briefly look at some important dimensionality reduction methods that we employ in this chapter. Broadly speaking, these methods fall into two groups: *Linear* methods and *non-linear methods*. Linear methods create embeddings as a linear transformation of the input data, for example by performing an analysis of the eigenvalues of certain matrices. Non-linear methods, by contrast, create intermediate representations of data, such as neighbourhood graphs, and calculate embeddings based on the properties of these representations.

Principal component analysis (PCA)

PCA is one of the first dimensionality reduction methods. Its conceptual simplicity makes it a common choice for analysing data; consequently, it is almost universally known in many different scientific fields.

To calculate a PCA, we first obtain a *sample covariance matrix* [216, pp. 119–123] of the input data. We then perform a spectral decomposition to transform the data into its eigensystem. By sorting eigenvalues in decreasing order, we obtain a simple cut-off criterion for determining when to 'stop' the embedding. There are numerous variants of PCA that are e.g. geared towards handling noisy data [219].

Multidimensional scaling (MDS)

MDS is based on the idea of adjusting a given configuration of points in some $\mathbb{R}^k$, the space of the embedding, until their distances are a good approximation of the distances in the original space. As for PCA, numerous variants of MDS exist [49, 224, 225].

MDS is one of the few methods that is capable of embedding data sets for which only pairwise distances between objects are known. This versatility makes MDS an attractive dimensionality reduction technique. We will use MDS in Chapter 8, for instance, to produce embeddings that are based on topological dissimilarities between data sets.

Isomap

Seen by some as an extension of MDS, Isomap is one of the first non-linear dimensionality reduction methods. It was developed by Tenenbaum et al. [359] to demonstrate that not all phenomena can be captured properly by linear dimensionality reduction methods. Isomap employs a neighbourhood graph (similar to the ones used in the calculation of persistent homology) to obtain an approximation of geodesic distances in the data. The matrix of geodesic distances is then embedded and visualized using MDS.

While there has been some controversy concerning the stability of the approach [21], Isomap continues to perform well for a variety of manifold data. Recent research concentrates on preserving properties of manifolds, such as curvature [340].

Locally linear embedding (LLE) & Hessian locally linear embedding (HLLE)

LLE was introduced by Roweis and Saul [322] to improve the handling of non-linear phenomena. It is based on the insight that data may be approximately linear, but only on local scales. The algorithm approximates the data set thus locally using linear patches, which are then put together to form a global embedding. The underlying auxiliary data structure of LLE is a graph of the k nearest neighbours of a point. Choosing suitable values for this parameter remains a challenge, requiring heuristics [324] and auxiliary visualizations.

HLLE is an improvement of the LLE algorithm that was developed by Donoho and Grimes [132]. The computational complexity of HLLE is lower because it employs a sparse Hessian matrix during the optimization phase in which the final positions of all data points are calculated. Although often referred to as *Hessian eigenmaps*, this thesis prefers using the term HLLE because it better reflects the similarity to LLE.

t-distributed stochastic neighbour embedding (t-SNE)

t-SNE was developed by van der Maaten and Hinton [256] with the aim of visualizing low-dimensional data. Internally, proximity between data points is modelled by constructing an appropriate t-distribution [386, p. 30] that fits the data. The *Kullback–Leibler divergence* [229] between the distribution on the original data and the distribution on the embedding is then minimized. Starting from an initial low-dimensional configuration, positions in the embedding are thus iteratively adjusted until a local optimum has been reached. t-SNE is capable of preserving both multi-scale structures and clusters in a data set [256]. Typically, it has a complexity of $\mathcal{O}\left(n^2\right)$, where n is the number of data points. In recent work, van der Maaten [255] described an optimization strategy that reduces the complexity to $\mathcal{O}(n \log n)$.

Random projections (RPs)

RPs are based on the Johnson–Lindenstrauss lemma [217] that states that small sets of points may be embedded in low-dimensional spaces such that their distances are nearly preserved. Baraniuk and Wakin [23] showed the applicability of this lemma for smooth manifolds.

The basic idea of RPs involves constructing a random projection matrix whose columns have unit lengths. In a strict sense, this matrix is not orthogonal—but it turns out that it is, as Bingham and Mannila [47] remark, 'sufficiently close to being orthogonal' to be of use as a projection. The computational simplicity makes RPs an attractive method for working with image and text data, for example [47].

Other approaches

There are several other approaches that we do not explicitly define in this section due to their similarity to other approaches. However, we briefly cite the relevant publications for reference purposes. Examples comprise linear methods such as *factor analysis* (FA) [100], but also recent non-linear methods based on neighbourhood graphs such as *local tangent space alignment* (LTSA) [405] or stochastic methods such as *stochastic proximity embedding* (SPE) [4].

Moreover, we neither discuss nor use methods based on diffusion processes [107] in the subsequent analysis. While these methods are used for a wide variety of applications, they require a fair amount of parameter tuning, which makes them somewhat unsuitable in the context of EDA.

To illustrate the issues that users of dimensionality reduction methods have to face, we show different embeddings of a manifold data set. We use an excerpt of the *MNIST handwritten digits data* [234]. Originally, the data set consists of 60,000 handwritten digits with a resolution of 28×28 pixels. Each picture may hence be considered to be a point in a 784-dimensional space. Since the complete data set is too large for some algorithms, we randomly extracted 100 digits of each class so that we obtain a more manageable data set with 1,000 digits. Using Tapkee, an efficient C++ library for dimensionality reduction by Lisitsyn et al. [249], we generated several embeddings that are depicted in Figure 7.1. Without the availability of additional information such as class labels, it is not easy to figure out whether an embedding is suitable or not. This issue becomes more relevant as many data sets created by scientific experiments do not necessarily contain—or even permit—a labelling.

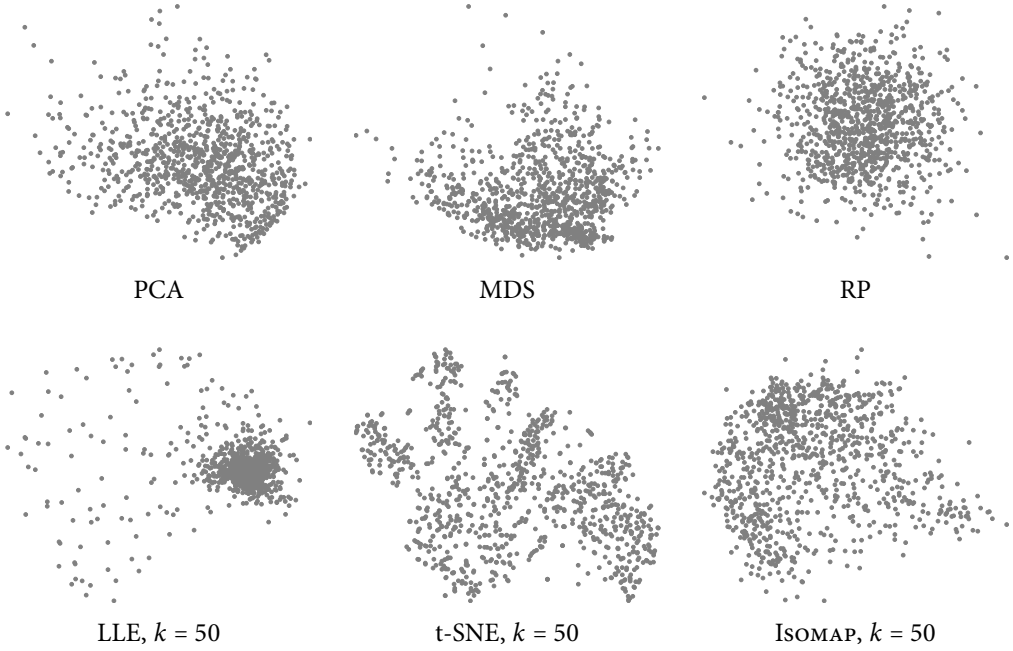


Figure 7.1: Example embeddings of the MNIST data. All dimensionality reduction methods present the data in very different manners. If no labels are given, how can we find out which of these embeddings are suitable? The k parameter refers to the number of neighbours that have been used in the corresponding neighbourhood graphs of some algorithms.

7.2 Quality measures

To judge the quality—or the suitability—of an embedding, users often employ *quality measures*. A quality measure compares some aspect, such as the distances, between the original data set and its embedding, yielding a single number that quantifies the error. Either through experience or through comparing different runs of an algorithm, users may thus judge whether a method is suitable for embedding the given data set.

In the following, we briefly review several common quality measures to get an idea of their properties. There are two large groups of quality measures: *Distance-based* measures (which may become unreliable in higher dimensions) and *rank-based* measures (which are prone to some instabilities if distances or neighbours are not sufficiently distinct). The literature dealing with distance-based measures is rather scattered. For rank-based quality measures, Lee and Verleysen [239] give an excellent overview. In general, distance-based measures are more stable against *small* changes in the embedding, while rank-based measures are more stable against *large* changes of the data, such as uniform scaling. For the following definitions, we shall assume that our data have a cardinality of n. We shall also transform the quality measures such that high values indicate regions of low quality—in essence, we transform the

quality measures to error measures. We will use d_{ij} to refer to distances in the original space and δ_{ij} to refer to distances in the low-dimensional embedding.

Definition 7.1 (Stress). Given an embedding, we compare the squared differences between the distances in the original high-dimensional space d_{ij} with the distances in the embedding δ_{ij}, using an appropriately-selected scaling factor:

$$Q_{\text{Stress}} := \sqrt{\frac{\sum_{i<j}\left(\mathrm{d}_{ij} - \delta_{ij}\right)^2}{\sum_{i<j}\delta_{ij}^2}} \tag{7.1}$$

Kruskal [224] defined stress as a loss function for solving an optimization problem. It is still in use today, as many modern multidimensional scaling algorithms aim to calculate a low-dimensional embedding that minimizes the stress. A disadvantage of stress is that it can be made arbitrarily large by scaling an embedding. If used as a quality measure, stress is biased towards algorithms that attempt to preserve the distances exactly as they occur in the high-dimensional space.

Definition 7.2 (Root-mean-square error). The RMSE measures the average squared difference between the distances in the original space and the embedding:

$$Q_{\text{RMSE}} := \sqrt{\frac{\sum_{j=1}^{n}\left(\mathrm{d}_{ij} - \delta_{ij}\right)^2}{n}} \tag{7.2}$$

This measure is also biased towards algorithms that preserve distances. It is often used to measure the quality of regression models. We will return to this aspect in Chapter 8.

Definition 7.3 (Local continuity meta-criterion). In order to avoid the dependence on distances, we may alternatively use the local neighbourhoods of points. In a good embedding, it is reasonable to assume that the k nearest neighbours of points are preserved to some extent. We shall subsequently refer to the indices of the k nearest neighbours of a point x in the high-dimensional space as $\mathrm{N}_k(x)$, whereas the neighbours in the embedding are denoted by $\mathcal{N}_k(x)$. Chen and Buja [96] describe the *local continuity meta-criterion* as:

$$Q_{\text{LCMC}} := \frac{1}{kn}\sum_{x}\left|\mathrm{N}_k(x) \cap \mathcal{N}_k(x)\right| - \frac{k}{n-1} \tag{7.3}$$

The criterion has the advantage of being insensitive to uniform scaling. It is normalized to an interval of $[0, 1]$, where higher values indicate a better quality. By varying the k parameter, the sensitivity for changes in neighbourhoods can be adjusted.

Definition 7.4 (Residual variance). This rank-based measure assumes that the original distances and the distances in the embedding are correlated. A perfect embedding should thus exhibit a perfect correlation of distances. We obtain a quality measure by calculating the complement of the explained variance between the distances. Writing R^2 for the square of Pearson's correlation coefficient, this leads to:

$$Q_{\text{Residual variance}} := 1 - R^2(\{\mathrm{d}_{i0}, \ldots, \mathrm{d}_{in}\}, \{\delta_{i0}, \ldots, \delta_{in}\}) \tag{7.4}$$

It would also be possible to use a different correlation measure in the previous equation. The correlation measure developed by Székely and Rizzo [353] is an interesting candidate.

Definition 7.5 (Spearman's rank correlation). By converting the distances d_{ij} and δ_{ij} to ranks r_{ij} and ρ_{ij}, respectively, this measure is more stable against outliers in the data and invariant to linear scaling. It involves calculating the Pearson correlation coefficient of the ranked distances:

$$Q_{\text{Spearman}} := 1 - 6\frac{\sum_{j=1}^{n}\left(\mathrm{r}_{ij} - \rho_{ij}\right)^2}{n(n^2-1)} \tag{7.5}$$

In contrast to the ordinary Pearson correlation coefficient, Spearman's rank correlation is capable of assessing monotonic relationships—linear and non-linear—between two variables.

Definition 7.6 (Mean relative rank error). The MRRE measures the mean amount of rank deviations using the k nearest neighbours of the point in both the original space and the embedded space. The calculation decomposes into two parts:

$$Q_{\text{MRRE, low–high}} := \frac{1}{C}\sum_{i=1}^{n}\sum_{j\in\mathcal{N}_k(x_i)}\frac{|\mathrm{r}_{ij} - \rho_{ij}|}{\rho_{ij}} \tag{7.6}$$

$$Q_{\text{MRRE, high–low}} := \frac{1}{C}\sum_{i=1}^{n}\sum_{j\in\mathrm{N}_k(x_i)}\frac{|\mathrm{r}_{ij} - \rho_{ij}|}{\mathrm{r}_{ij}} \tag{7.7}$$

The normalization factor C ensures that the errors remain bounded in $[0,1]$. It is defined as a worst-case assumption of not preserving any neighbours, i.e.

$$C := n\sum_{l=1}^{k}\frac{|2l - n - 1|}{l}, \tag{7.8}$$

where the value depends on the number of neighbours k that are used to calculate the local neighbourhoods. The MRRE was introduced by Lee and Verleysen [239] to penalize two common errors in embeddings, namely very distant points that *intrude* into the k nearest neighbours of a point, as well as very close points that *extrude* from such a neighbourhood.

Algorithm	Q_{RMSE}	Q_{Stress}	Q_{MRRE}	Q_{Spearman}	$Q_{\text{Residual variance}}$
Isomap	2247.59	0.999	0.0032	0.56	0.82
LLE	**0.75**	280.291	0.0033	0.76	0.95
MDS	15 008.10	0.998	0.0032	**0.46**	**0.68**
PCA	688.45	0.998	0.0033	0.70	0.94
RP	85.73	**0.931**	0.0033	0.90	0.99
t-SNE	19.21	0.984	**0.0025**	0.63	0.86

Table 7.1: Selected quality measures for some embeddings of the MNIST data. The best value in every column has been marked. The quality measures sometimes differ by several orders of magnitude. This example also demonstrates that there is no 'clear' winner; only few measures agree in their assessment of the quality of an embedding.

Definition 7.7 (Neighbourhood loss). Since local neighbourhoods play an integral part in the perception of an embedding, they should ideally be respected to some extent. The *neighbourhood loss* measure quantifies how many neighbours, on average, are lost during the embedding process. The measure is fully agnostic to distances and requires an enumeration of the k nearest neighbours of a point both in the original data and the embedding, which makes it highly-scalable. Denoting the original neighbourhood of the point by $\mathrm{N}_k(i)$ and the embedded neighbourhood of the point by $\mathcal{N}_k(i)$, we have:

$$Q_{\text{Neighbourhood loss}} := 1 - \frac{\left|\mathrm{N}_k(i) \cap \mathcal{N}_k(i)\right|}{k} \tag{7.9}$$

This measure only attains values in $[0, 1]$, as well. A value of 1 indicates that all k nearest neighbours change during the embedding.

A comparison of different quality measures

Table 7.1 shows some selected quality measures for the embeddings depicted in Figure 7.1. The best value has been marked in each column. A closer observation indicates numerous issues with the different measures, though.

First, there are often no clear distinctions between different values. For example, the perceptual differences between several embeddings are large, while their Q_{Stress} values are very similar except for the obvious outlier of LLE. Likewise, many of the Q_{MRRE} values hardly differ, even though the embeddings appear to highlight very different aspects of the data. Furthermore, the unbounded scales of some of the measures make it hard to compare different embeddings and establish a meaningful baseline. The stress value of the t-SNE embedding, for example, appears to be very low in comparison with the stress values of the other embeddings. However, the embedding calculated with LLE manages to achieve an even smaller stress value.

It is not immediately clear whether the difference between the two embeddings makes them 'more related' to each other. The visualizations of the embeddings, shown in Figure 7.1, indicate that the embeddings are at least perceptually extremely different.

Since there are even more quality measures for various purposes, how can we practically apply them to a set of embeddings? A common approach is to obtain multiple embeddings of data, generated by different dimensionality reduction methods, and calculate the desired quality measure on them. In this case, the best embedding is the one that minimizes the given quality measure. While this approach has its merits, it assumes that the quality measure is not biased (in the sense of *stress*, for example). Furthermore, the quality measure needs to be robust against perturbations of the data as well as sufficiently expressive, meaning that it should be able to separate 'bad' embeddings from 'good' ones.

Perceptual aspects

Existing quality measures are theoretically sound, but do not exploit how humans perceive an embedding. Tatu et al. [357] showed that when users look at scatterplots of embeddings, they are naturally drawn towards measurable properties of these plots, such as their density. In case data are expected to exhibit clear clusters, Sedlmair et al. [331] proved that 2D scatterplots are sufficient to detect class separation. Finally, Lewis et al. [242] conducted an experimental study of whether humans agree in their assessment of the quality of embeddings. The results indicate that expert users, who are aware of the internal model of an algorithm, are consistent in their ratings and well capable to assess the quality of an embedding. For novice users, on the other hand, no such guarantees hold—in fact, the assessment of novice users was even slightly negatively correlated with the known quality of an embedding. Moreover, novices appear to consider embeddings with little clustering behaviour to be better than embeddings with pronounced clusters.

If we want to evaluate dimensionality reduction methods under these aspects, we need a way to quantify how well certain intrinsic properties–such as the density—of our data are preserved in different embeddings.

Towards a holistic description

In the following, we will quantify this preservation from two different viewpoints. First, we will evaluate the agreement of quality measures on a given embedding. This permits us to detect regions in which different properties of a data set are not preserved. Second, we will derive a workflow that uses a scalar function on the data set in order to quantify how well its properties are preserved globally and locally. Together, these two approaches yield a holistic description of embeddings of multivariate data sets.

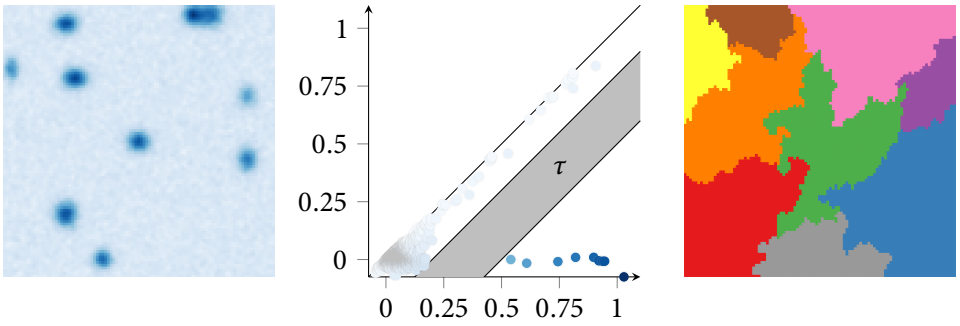
7.3 Agreement analysis

In Table 7.1, we have already seen that some quality measures agree in their judgement of a data set. How can we quantify this agreement more formally, especially in light of the scale differences of some measures? First, we observe that most of the measures presented in the sections above afford a *localized calculation*. For example, we could calculate Q_{Stress} locally by summing only the distances to all other points of a given point. As a result of these localized calculations, we obtain a scalar function on the high-dimensional original data. These scalar functions could conceivably be visualized on the embedding itself [17, 271], permitting users to find out whether certain parts of the embedding are more trustworthy than others. However, as we shall also see in the subsequent chapters, the *visual* comparison between different scalar functions is tedious and error-prone. Hence, this approach is not wide-spread. Since quality measures may be calculated efficiently—in almost all cases more efficiently than the embedding itself—a large amount of information is potentially being wasted by not considering them.

Persistent homology permits us to take a different viewpoint here. By treating the quality measure values as a *scalar field* on the data, we may partition an embedding into regions of uniform behaviour (in a sense that will become more clear later on) and compare those partitions among each other. The intuition behind this is that if two quality measures result in similar partitions, we may assume that their behaviour is similar—regardless of their actual values, they exhibit the same pattern of errors. In the following, we shall introduce a novel algorithm for comparing these partitions among each other. While the method presented here is more coarse than comparing the values of the measures directly, it provides a rapid high-level overview of multiple aspects of a single embedding. This overview turns out to be sufficient for many analysis tasks.

Decomposing a scalar field

In Chapter 5, Section 5.3, p. 90 ff., we have already encountered a clustering algorithm based on the idea of mode-seeking [99]. We may reformulate this algorithm and use it to decompose our original data into disjoint subsets. These subsets have the property that one arrives at the same local 'peak' when following the discrete gradient of the function. The difference to the previous usage of the algorithm is that we use the localized quality measure function as the underlying function of the algorithm. As a consequence, the data set will be decomposed in terms of regions with a similar behaviour of errors. Figure 7.2 illustrates this.



Noisy scalar field | Persistence diagram | Decomposition

Figure 7.2: An illustration of the decomposition algorithm. Starting from a noisy scalar field, we use the previously-defined heuristic and decomposition algorithm to obtain a persistence diagram. The persistence diagram exhibits a clear separation between signal and noise. Hence, the empty region from which we may choose τ is very large. All examples employ a continuous colour map that uses darker colours to indicate higher values.

Parameter selection

Recalling the original clustering algorithm, we require two parameters. First, ϵ controls the coarseness of the topological approximation. We have already encountered good heuristics in Section 5.4, p. 96 ff., for selecting it. Second, we have the τ parameter that controls the required 'prominence' of a peak in the data. Previously, we let the user select this parameter by means of a persistence diagram. Here, we want to present another heuristic that is motivated by a theoretical result of Chazal et al. [91, Theorem 4.8], which states that significant peaks can be extracted from the persistence diagram if the diagram does not contain any points within a band of a certain width that starts at the diagonal. In other words, if the distinction between 'topological signal' and 'topological noise' is large enough, all relevant peaks can be extracted. The theorem makes several assumptions about the structure and the sampling conditions of the input data, which are unavailable to us when dealing with real-word data sets.

We thus propose a threshold selection that is based on the idea behind the theorem. It involves finding the largest empty region parallel to the diagonal that we can draw into the persistence diagram. If this empty region is sufficiently pronounced—which we can measure by evaluating *all* possible empty regions in the diagram at once—we can pick any point within the region to obtain an admissible value for the threshold parameter τ.

To find the largest empty region, we transform the coordinate system of the persistence diagram via a rotation by $\pi/4$. Thus, the diagonal becomes the abscissa of the new coordinate system. We now sweep over all points by descending y-coordinates and keep track of the vertical distance between subsequent points. The largest vertical distance is the width of the largest empty region. Its corresponding y-coordinate then yields the desired value for τ.

We reject this value if it is not outside 1.5 times the interquartile range of all widths that we encountered. This ensures that only an extremely outlying width is being considered a valid value for τ. Figure 7.2 illustrates the calculation. In this case, our algorithm suggests a value of $\tau \approx 0.33$. The decomposition would remain stable for $\tau \in [0.18, 0.48]$.

Measuring similarity between decompositions

Having obtained a decomposition of the input scalar fields by means of the persistence diagram, we now want to assess its similarity to other decompositions. To this end, we use the *Jaccard index* [402, p. 435] from data mining. Given regions A and B, which we assume to be described by a set of vertex indices, their similarity is defined as

$$\mathrm{J}(A, B) := \frac{|A \cap B|}{|A \cup B|}, \tag{7.10}$$

with $\mathrm{J}(A, B) \in [0, 1]$. The maximum value of the Jaccard index indicates that the two regions are equal. In contrast to other indices that measure the similarity of partitions, the Jaccard index has the advantage of being defined for partitions with different sizes. To use the index as a similarity measure, we define an *assignment problem* [226] that quantifies the global similarity between two decompositions. The cost for matching two regions A and B is

$$\mathrm{cost}(A, B) := 1 - \mathrm{J}(A, B), \tag{7.11}$$

meaning that we want to penalize regions that do not overlap. To make a perfect matching possible, we also include 'dummy regions' that account for differences in the cardinality of the sets. The cost for matching against these regions is set to 1. That way, regions that have no common intersection are rather matched to their dummy region than to a real region. The total cost of this assignment problem shows how much the two decompositions differ.

By solving the assignment problem for all pairs of decompositions, we obtain a matrix of costs. Next, we calculate a one-dimensional PCA on this matrix in order to get a linear ordering of the scalar fields that reflects their relative distances. In this ordering, similar scalar fields are being placed in close proximity to each other. This arrangement permits us to quickly read off which properties of an embedding are most likely retained to a similar extent.

Local agreement visualization

To permit the assessment of quality on a local scale, we require a reference scalar field. This field can be chosen by users depending on the quality measure that is to be considered most important, e.g. stress. We may now solve the assignment problem for each remaining scalar

field and analyse the matching costs with respect to the reference field. This procedure assigns each region in the reference field a set of costs. We visualize these costs on the embedding using three colours. Blue ● indicates low costs, yellow ● indicates medium costs, and red ● indicates high costs. Each colour comprises one third of the range of values. Generally, blue ● regions indicate that a region in the reference field is in good agreement with the remaining scalar fields. The distribution of errors in such a region is thus very similar in all fields. By contrast, yellow ● regions highlight mild differences between the different fields, whereas red ● regions indicate that parts of a field are severely mismatched. These regions tend to be unique among all scalar fields and point out erroneous parts of an embedding. We shall see examples in the next section.

In the optimal case, all other scalar fields result in the same decomposition as the reference field. The assignment problem hence always results in a perfect matching and the complete embedding is shown in blue—in this case, the visualization does not highlight any particular region. In practice, this only occurs rarely.

7.4 Results of agreement analysis

In the following, we shall take a look at two multivariate data sets and analyse the properties of their embeddings. We assume that the user has selected a quality measure that is to be minimized on a global scale. Our task is now to find out whether *other* measures are behaving similarly on the data as this will shed some light on problematic regions.

Pre-processing

As a data pre-processing step, we normalize the values of each quality measure to $[0,1]$, where 0 represents the highest quality (no errors) and 1 represents the lowest quality (maximum amount of errors). This step is only required to ensure that the scales of the different persistence diagrams calculated by our method are comparable with each other. Alternatively, it would be possible to use *ranked values* and compare persistence diagrams based on their *indices*—this approach was favoured by Zomorodian and Carlsson [409]. Its disadvantage is the loss of scale information; the difference in indices could become arbitrarily large. Hence, we prefer the pre-processing step here. This issue could also be solved by adjusting the distance calculations between persistence diagrams to measure only differences in topological feature aggregation.

Colours

In the following sections, we will encounter multiple embeddings with an associated scalar value that represents a given quality measure. Each of these visualizations uses a continuous

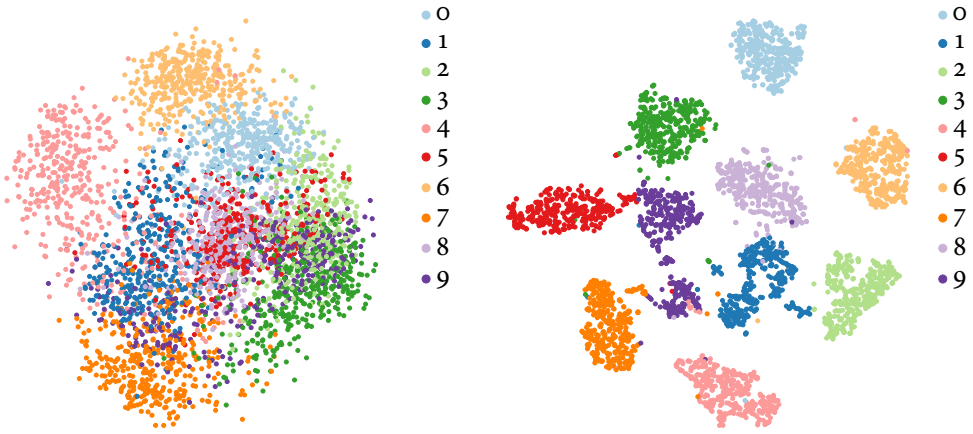


Figure 7.3: PCA and t-SNE embedding of the handwritten digits data. The class separation is significantly better for t-SNE, but this comes at the expense of a higher runtime. Regions between digits, for example between '7' and '9', are interesting, as they contain digits that are written imprecisely such that they resemble another digit.

colour map that represents high values with dark colours. Since these representations are meant to be illustrative only, we refrain from including additional legends in the plots. In the local agreement visualizations, we use the previously-introduced categorical colour scheme for indicating the matching costs (blue ● for low matching costs, yellow ● for medium matching costs, and red ● for high matching costs).

7.4.1 Handwritten digits

The *Optical Recognition of Handwritten Digits* data set, which we refer to as handwritten digits data, is provided by the UCI Machine Learning Repository [247]. It contains 5,260 instances of 64-dimensional feature vectors that describe handwritten digits of different writers. The feature vectors are known to lie on a lower-dimensional manifold [200], making the data set an ideal candidate for manifold-based dimensionality reduction algorithms. In contrast to the MNIST data that we used as an example earlier, this data set is not prohibitively large for some algorithms while exhibiting a similar internal structure.

In a first experiment, we compare the behaviour of different quality measures between PCA and t-SNE. Figure 7.3 shows the two embeddings. We observe that the PCA embedding exhibits many overlaps, especially between digits '5', '8', and '9'. The t-SNE embedding, by contrast, shows a good separation between the individual classes. Why should we thus bother with the PCA embedding? One possible answer lies in the scaling behaviour of the two algorithms. Calculating the PCA embedding takes 0.76 s, while the t-SNE embedding



(a) MRRE (b) MRRE decomposition (c) Rank correlation



(d) Residual variance (e) RMSE (f) Stress

Figure 7.4: Quality measures for the PCA embedding of the handwritten digits data. Except for the MRRE measure, all measures show that errors concentrate in the core region of the data and decrease when going to more outlying regions. As a consequence, only the first measure results in a decomposition with more than one peak. Our colour scheme uses darker colours to indicate larger errors.

requires 17.64 s on the same system. Especially if vast amounts of data need to be compressed, visualized, and analysed, users may well prefer a fast answer that is *somewhat* correct to a *perfect* answer that takes a prohibitive amount of time. Furthermore, PCA and MDS are among the most intuitive dimensionality reduction algorithms and are often the first choice for most users [329]. It is thus important to be aware of their capabilities and limitations.

PCA

Most of the quality measures result in the same decomposition when analysed by our algorithm. In Figure 7.4, we only show an excerpt. Ignoring the *mean relative rank error* (MRRE) for a moment, we observe that local quality is characterized by very low values in the core region of the embedding, where we also experience the largest amount of overlaps between classes of digits. Quality starts to increase when going to the more outlying points. The decomposition of

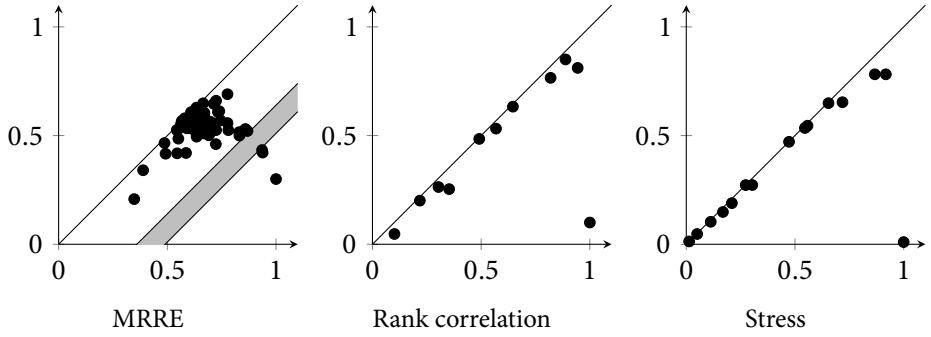


Figure 7.5: Persistence diagrams of the handwritten digits PCA embedding. Only the MRRE measure exhibits more than one significant peak.

all these measures results in a single peak of large persistence, located in the dense core region. While we perceive other peaks, their persistence values are not sufficiently high. Figure 7.5 depicts the persistence diagrams of the corresponding measures. The diagrams indicate that those fields do not exhibit clearly-defined peaks—except for the MRRE decomposition. For this quality measure, we observe more than one prominent peak in the embedding, resulting in a more pronounced decomposition depicted in Figure 7.4b. By contrast, errors in the 'Neighbourhood loss' measure are large everywhere because smaller neighbourhoods (less than twenty neighbours) get distorted in the PCA embedding. When comparing the errors with the class labels, we see that they concentrate on the core region, containing the digits '5', '8', and '9'. Apparently, these classes cannot be separated well by PCA.

In summary, this embedding is a typical example of a compromise solution. It can be generated quickly at the expense of visual quality—all quality measures exhibit large errors. Those errors accumulate in a single region, though. Said region is not critical to understanding the data, so the PCA embedding is sufficiently good to get a quick overview of the data. We refrain from showing the relative distances between the decompositions. Since all decompositions are similar except for one, we will not obtain any additional information here. We may demonstrate the visualization of local quality, however. Figure 7.6 shows two visualizations with different reference quality measures. Of course, these visualizations are not highly-informative in this case. From the point of view of most of the measures, there is no disagreement about the errors in the data because all measures merely detect a single large peak in the core of the data. Only when using the MRRE measure as a reference do we see some disagreement about the more outlying regions of the embedding. This is consistent with the assessment of the other measures, though—all measures exhibit large errors in the core region of the data, but only MRRE shows large errors in other regions, as well.

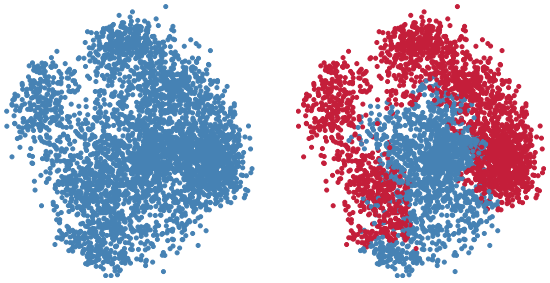


Figure 7.6: Local agreement for the PCA embedding of the handwritten digits data. We can see that compared to stress, for example, all measures appear to agree. This is consistent with their decompositions. If we use the MRRE measure as a reference, however, we can see that there is some disagreement in the outlying areas of the embedding. This is indicated by the high matching costs ● for two regions, while the remaining region has low matching costs ●.

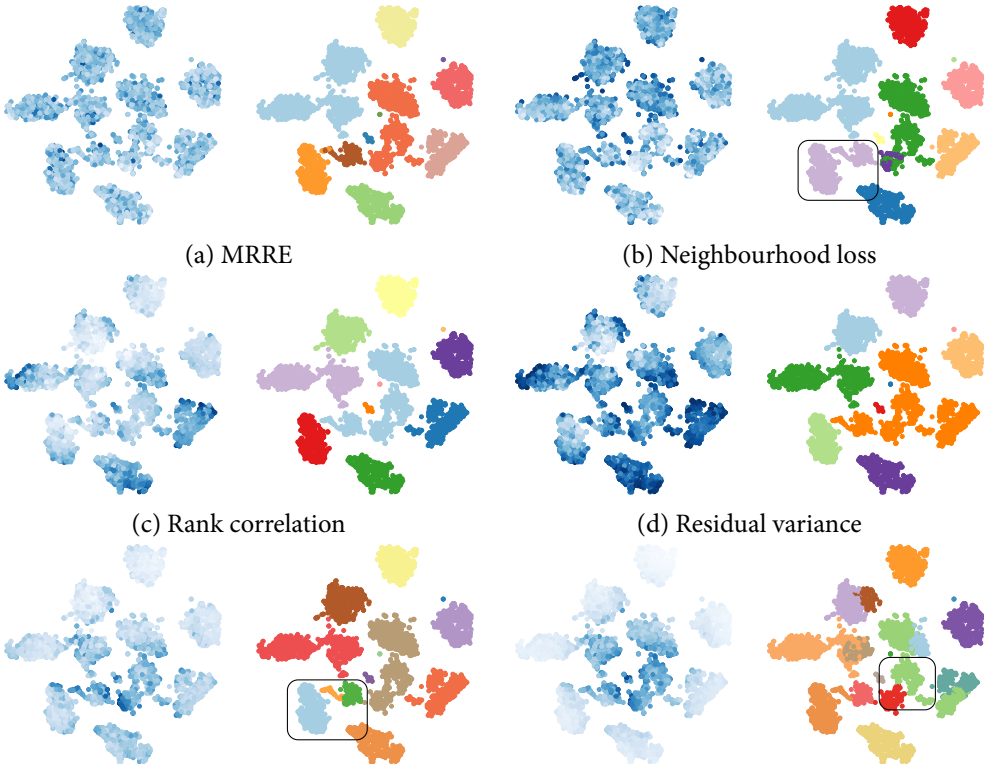
t-SNE

The t-SNE embedding exhibits a more pronounced distribution of errors, leading to a more detailed region-based quality assessment. Figure 7.7 on p. 164 depicts visualizations of the individual measures. The 'Rank correlation' measure, shown in Figure 7.7c, for example, highlights errors in some of the more outlying clusters of the individual digits. Figure 7.7e indicates that RMSE, on the other hand, features some errors in the 'core' of the embedding.

Our connectivity approximation ensures that the Rips graph $\mathcal{R}_\epsilon$ of the data set only has a single connected component. Nonetheless, some quality measures such as 'Residual variance' result in a single peak per class, i.e. per digit. This behaviour implies that errors are distributed evenly among the classes. We now shortly discuss some quality measures, their decompositions, and the implications for the resulting embedding.

Neighbourhood loss The 'Neighbourhood loss' measure is an exception—the highlighted region of the embedding is not split up into smaller subregions. It contains only some instances of '9' that are more similar to '7' than to the other instances of '9'. When being placed in the immediate vicinity of the cluster of '7', their local neighbourhoods remain almost unchanged so that they do not contribute to any errors. Figure 7.8a demonstrates this issue by showing the raw data, i.e. the images of the digits as they were originally drawn.

Root-mean-square error (RMSE) For the RMSE measure, an additional split is introduced in the previously-encountered region; see Figure 7.7e for details. This split is justified, because following the small excerpt of the embedding shown by Figure 7.8a, we can see that t-SNE creates a perceptual similarity between instances of '7' and '9' that is not justified by

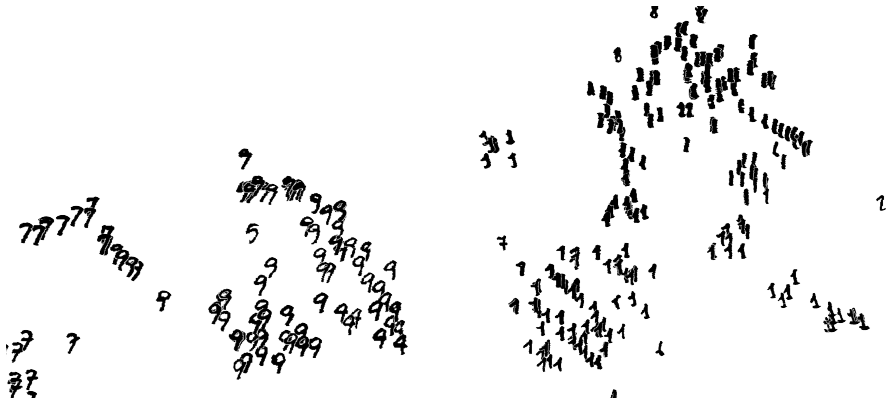


(a) MRRE (b) Neighbourhood loss

(c) Rank correlation (d) Residual variance

(e) RMSE (f) Stress

Figure 7.7: Quality measures for the t-SNE embedding of the handwritten digits data. In contrast to the PCA embedding, the differences between the quality measures are more pronounced here. The sequential colour map uses darker colours to indicate higher errors. We refrain from showing labels in order to keep the visualization illustrative.



(a) Separation between '7' and '9'. (b) Different degrees of similarity for '1'.

Figure 7.8: Example regions in the t-SNE embedding. When written imprecisely, some digits start to resemble other digits. Perceptually, their placement in the embedding is not always correct.

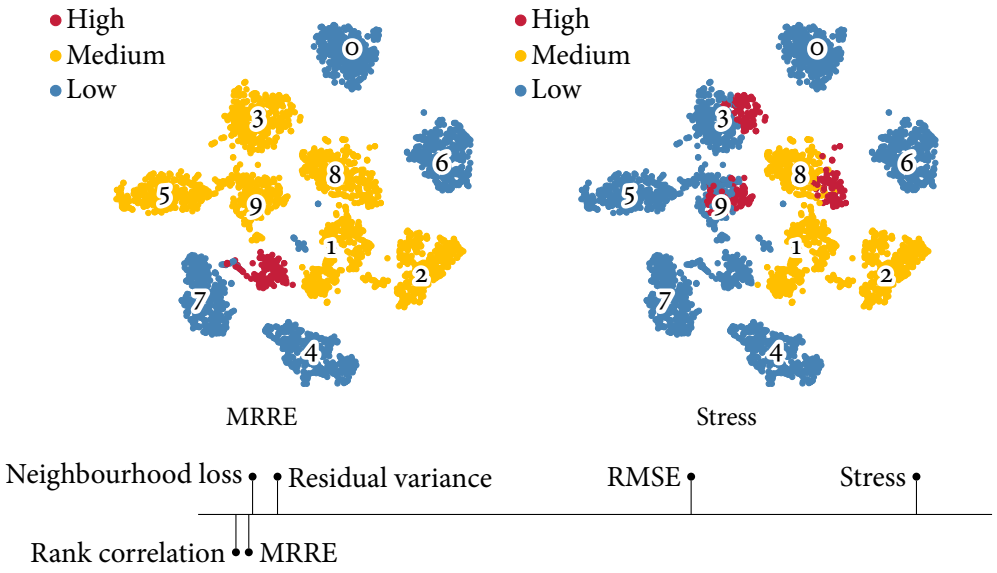


Figure 7.9: Local agreement for the t-SNE embedding of the handwritten digits data. When compared to the MRRE measure, we can see that the stress measure disagrees in different regions of the data. In particular, there is no consensus about digits that may be confused with one another, such as '3' or '8'. However, in the relative agreement diagram, we can see that there is a large amount of agreement between four of the measures, indicating that their error distributions are similar. The legend in each plot shows the matching costs. Lower values are desirable because they represent regions in which multiple quality measures agree.

their internal representation. Hence, the RMSE measure detects a prominent peak, which results in the entire region being split into multiple parts. RMSE thus shows that the t-SNE embedding distorts the distances for these points in favour of an improved global separation.

Stress Another interesting and fine-grained decomposition is given by the stress measure. The highlighted region in Figure 7.7f corresponds to separate parts of the region containing the digit '1'. The high stress values are caused by large distances to instances of the digit '7' in the embedding. The apparent separation between the digit classes in the embedding does not reflect the fact that the corresponding handwritten digits are extremely similar. Figure 7.8b depicts an excerpt of these digits. We can see that the lower part of this region should rather be placed next to the '7' region because the differences are minuscule.

Agreement visualizations We now use MRRE as a reference measure and calculate the *relative agreement visualization*. Figure 7.9 shows that 'Neighbourhood loss', 'MRRE', 'Rank correlation', and 'Residual variance' appear to agree in their quality assessment on the data. However, we do not perceive this as easily in the direct visualizations of the measures, which

are depicted in Figure 7.7. This issue demonstrates that the coarse—but also stable—assessment obtained using the topology-based approach is advantageous. In a similar manner, the *region agreement visualization*, as shown in Figure 7.9, indicates problematic parts of an embedding with respect to a selected quality measure. In this case, we can see that when we use 'MRRE' as a reference measure, the largest disagreements occur for parts of the classes '1', '2', '3', '5', '8', and '9'. When using stress as a reference measure, though, only parts of the class '3' are highlighted as being in disagreement, for instance. The errors in this region turn out to be very small, but several other measures, such as 'Residual variance', detect higher errors. This may help re-evaluate the assessment of a single quality measure.

Conclusion For these data, an analyst might conclude that the t-SNE embedding is to be preferred because at least three quality measures agree on a distribution of errors. The regions of disagreement concern predominantly those digits that are easily confused. The loss of quality introduced is thus acceptable since the remaining digits have been embedded with very low distortions overall.

7.4.2 Concrete compressive strength

Having seen that the method works for data sets with a pronounced manifold shape, we now analyse a data set with no such structural guarantees. The *concrete compressive strength* data set from the UCI Machine Learning Repository [247] contains 1,030 mixtures consisting of 8 different concrete compounds. Table 7.2 describes the attributes and their types. These data were first collected and analysed by Yeh [401] in the context of predicting the compressive strength of different cement mixtures. Structural engineers consider such predictions to be important because they help them create more resilient concrete mixture with potentially lower costs. One of the results of the analysis by Yeh was that there is a complex relationship between the measured attributes and the compressive strength of a mixture.

✦

Previous research by Lee et al. [237] showed that MDS yields a suitable embedding of the data. This conclusion was reached by manually inspecting different groups of concrete mixtures in the data. We now want to find out whether we are able to reach a similar conclusion based on the agreement of different quality measures. Figure 7.10 shows the MDS embedding of the data. If we add colours according to the compressive strength of the mixture, we see that some mixtures with a high compressive strength appear to be concentrated in the upper part of the embedding.

Attribute	Unit	Type
Cement	$\mathrm{kg\,m^{-3}}$	Continuous
Blast furnace slag	$\mathrm{kg\,m^{-3}}$	Continuous
Fly ash	$\mathrm{kg\,m^{-3}}$	Continuous
Water	$\mathrm{kg\,m^{-3}}$	Continuous
Superplasticizer	$\mathrm{kg\,m^{-3}}$	Continuous
Coarse aggregate	$\mathrm{kg\,m^{-3}}$	Continuous
Fine aggregate	$\mathrm{kg\,m^{-3}}$	Continuous
Age	d	Discrete
Compressive strength	MPa	Continuous

Table 7.2: Attributes in the 'Concrete compressive strength' data set. The last attribute is typically considered an *output variable* because it depends on the values of the other attributes.

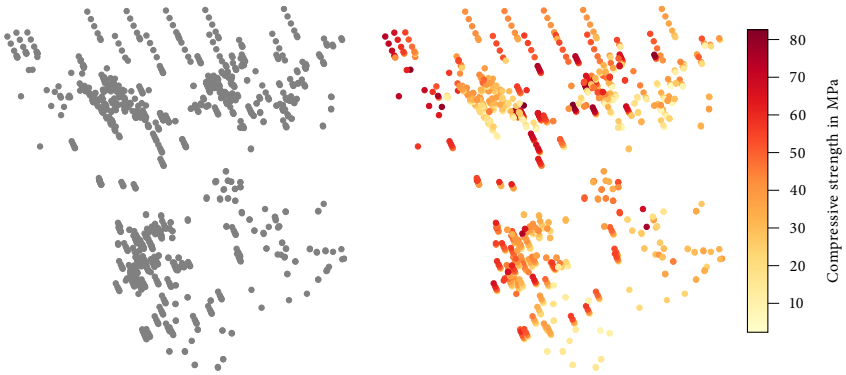


Figure 7.10: MDS embedding of the 'Concrete compressive strength' data. On the left, we show the raw embedding without any additional information. The linear structures in the upper part of the embedding are remarkable. On the right, we added colours according to the compressive strength of the mixture. Higher values are preferable.

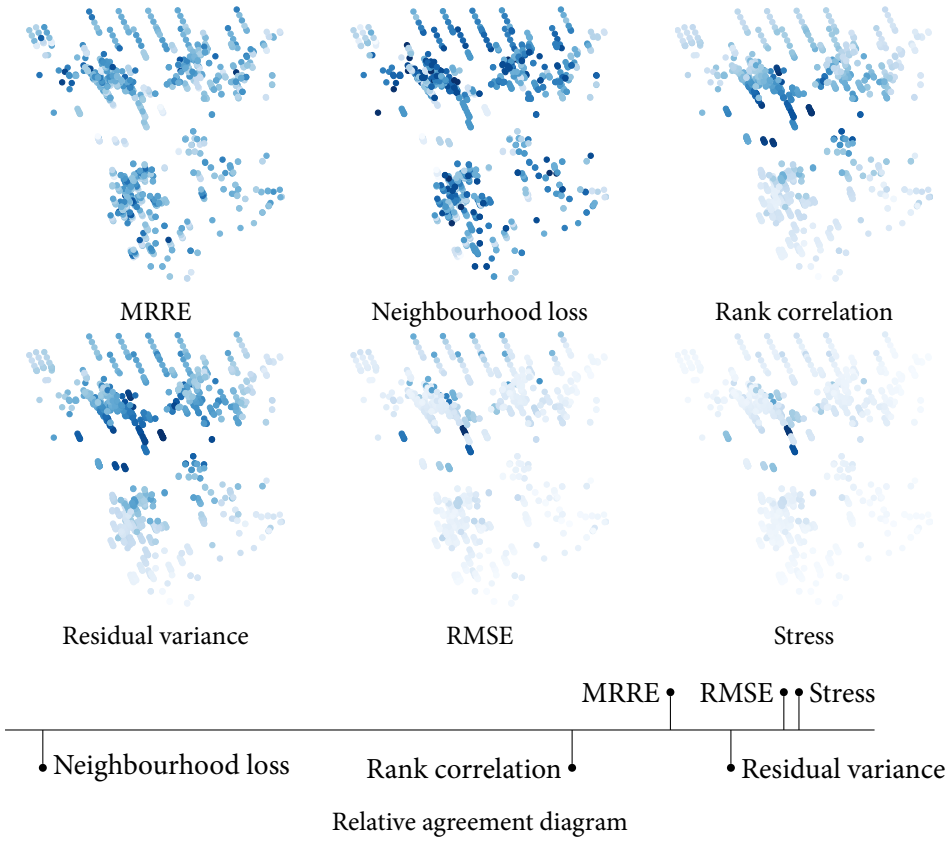


Figure 7.11: Quality measures for the MDS embedding of the 'Concrete compressive strength' data. The colour map uses darker colours to indicate larger errors. Several measures exhibit errors mostly in the upper part of the data, while others show that small errors are not restricted to any particular region.

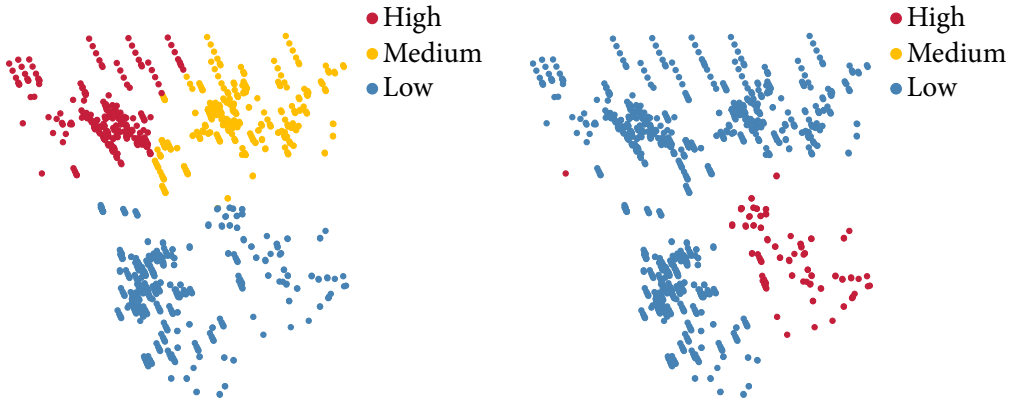


Figure 7.12: Local agreement for the MDS embedding of the 'Concrete compressive strength' data. We observe that many measures disagree with the MRRE measure, in particular with respect to the linear structures within the embedding.

Analysis of linear structures

Figure 7.11 depicts the 'raw' quality measures on the data. The relative agreement diagram in the bottom shows that the 'Neighbourhood loss' measure is extremely dissimilar to the remaining measures. RMSE and stress are considered to be very similar. The remaining measures do not appear to exhibit a clear grouping structure. Since the linear structures in the upper part of the embedding are a striking feature, we select the 'Neighbourhood loss' measure as a reference quality measure. Assuming that these structures are not a particular artefact of the embedding, they should exhibit low errors in the local neighbourhood-based criteria, such as MRRE and 'Neighbourhood loss'. Figure 7.12 shows the local region-based agreement with two reference quality measures. We can see that MRRE disagrees with the remaining measures. Moreover, those disagreements focus largely on the upper regions of the embedding. Since MRRE is one of the few measures exhibiting low–medium errors in this region, we may have some reservations about the validity of these structures. The 'Neighbourhood loss' measure, on the other hand, agrees with most measures in most of the regions. Here, the lower-right region is shown as being in disagreement. 'Neighbourhood loss' is the only measure that decomposes the lower part of the embedding into more than two regions. This indicates that there are some high-persistent errors among the mixtures depicted in this area. The other measures are incapable of detecting them.

Summary

The agreement analysis shows that there is no exact consensus about the distribution of errors in the embedding. Linear structures—a striking feature in the embedding—are assessed very

differently by the quality measures, which lead us to question their veracity. Hence, there are numerous reasons to suggest that the previous analysis by Lee et al. [237] does not account for misrepresentations and structural artefacts in the MDS embedding.

This leads us to the question of how to assess the *overall suitability* of the MDS embedding. The agreement analysis only gives us information about the 'relative behaviour' of quality measures. The remainder of this chapter thus describes a novel workflow that permits us to rate different embeddings with respect to how well they preserve certain properties of a data set. In contrast to the previously-encountered quality measures, this evaluation will have well-defined lower and upper bounds.

7.5 Data descriptors

So far, we have seen that existing quality measures are not sufficiently expressive for visualizing high-dimensional data. The novel agreement analysis, which we detailed in the previous sections, can be used to mitigate this issue to some extent, but it only permits the visualization of properties of a *single* embedding with respect to *multiple* quality measures. In the remaining part of this chapter, we describe the dual approach to agreement analysis—we will analyse the suitability of *multiple* embeddings by means of a *single* property. This approach is motivated by the perceptual usage patterns of dimensionality reduction users. Suppose we describe salient properties of our data, e.g. density, by means of a scalar function. Furthermore, assume that such a function is computable in the original high-dimensional space as well as in an embedding. Using persistent homology, we may then quantify the differences between the original data and the embedding in a meaningful and stable manner. We have already seen in Chapter 4, particularly in Section 4.6.3, p. 75 ff., that topological distances have advantageous properties in comparison to more common function space distances such as L_p or L_∞.



In the following, we will define some scalar functions that are specifically geared towards the analysis of high-dimensional data sets. Using auxiliary descriptor functions to simplify working with complex objects is a common strategy in the context of shape analysis [42, 44]. Here, the tool of choice is the family of *heat kernel signatures* [57, 352], which are capable of characterizing shapes up to isometry [181].

Definition 7.8 (Data descriptor). Given data $\mathbb{X} \subseteq \mathbb{R}^d$, we call a function $f\colon \mathbb{X} \to \mathbb{R}$ a *data descriptor* if it measures a salient property of $\mathbb{X}$, such as its density, and permits a calculation on an embedding $\mathbb{Y} \subseteq \mathbb{R}^l$, with $l < d$.

The subsequent sections contain some examples of data descriptors that we consider to be useful in the context of multivariate data analysis. We do not claim that this is an exhaustive listing; in fact, we consider the enumeration of further salient measurable properties of multivariate data to be a promising avenue for future research.

Density

We already discussed several density estimation approaches in Section 5.3.1, starting on p. 91. The *distance to a measure* density estimator [46, 89] is a prime example of a data descriptor function, as its calculation works in spaces of arbitrary dimensionality. Furthermore, density is known as a salient property in both data mining [160, 199] and visualization [286].

Eccentricity

The eccentricity of a data point is a measure of its centrality with respect to the other data points. This concept is borrowed from social network analysis [169, 170], where it is known to quantify structural information about a network. Eccentricity is a convenient concept because it does not require the definition of a central point, such as a barycentre or a medoid. Hence, the measure is very robust. We have

$$f_{\text{Eccentricity}}(x) := \frac{\sum_{y \neq x} \text{dist}(x, y)^2}{n - 1}, \tag{7.12}$$

where n is the number of data points and $\text{dist}(\cdot,\cdot)$ denotes a distance. High values in this measure indicate data points that are located more on the periphery of a data set. Low values, on the other hand, correspond to data points that are more central. The utility of this eccentricity measure has since been analysed by Lum et al. [253] and Carlsson [67]. A natural generalization with higher exponents in the numerator of the previous equation exists but does not provide further insights.

Local linearity

Linear structures commonly occur in scientific data sets. The idea that data may exhibit linearity at a local level is the central concept of the LLE algorithm [322], for example. In order to judge the linearity of a neighbourhood of points, we may use the following strategy. First, we enumerate the k nearest neighbours of our input point x. We then build a *covariance matrix* of dimensions $k \times k$, i.e.

$$\Sigma := \frac{1}{k} \sum_{i=1}^{k} (x_i - \overline{x})(x_i - \overline{x})^T, \tag{7.13}$$

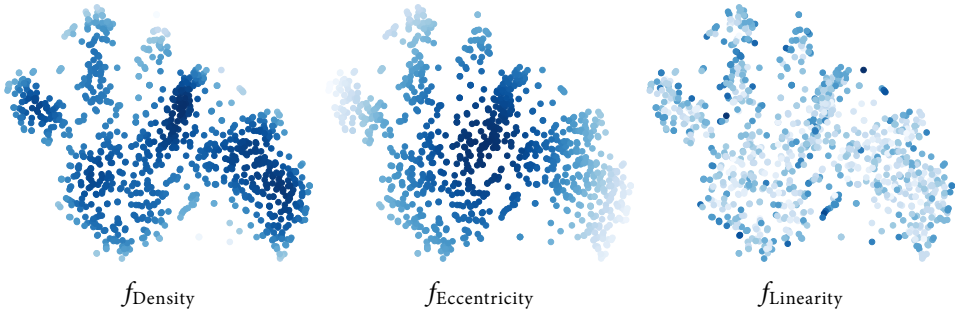


Figure 7.13: Selected data descriptors for an embedding of the MNIST data. High values are shown using darker colours. Note the subtle differences between the density data descriptor and the eccentricity data descriptor.

representing the spread around x. In the previous equation, $\overline{x}$ refers to the sample mean of x. Since Σ is a real-valued symmetric matrix, it affords a diagonalization. Hence, we can decompose the matrix into

$$\Sigma = PDP^{-1}, \tag{7.14}$$

where D is a diagonal matrix containing the eigenvalues $\lambda_1, \ldots, \lambda_k$ and P is a transformation matrix whose columns contain the eigenvectors of Σ. Without loss of generality, we require that $\lambda_1 \geq \ldots \lambda_k$. We now use

$$f_{\text{Linearity}}(x) := \frac{\lambda_1}{\lambda_1 + \cdots + \lambda_k} \in [0, 1] \tag{7.15}$$

as our data descriptor. Equation 7.15 measures how much of the variance of the data is explained if we only use a linear subspace, spanned by the first eigenvector (that corresponds to the largest eigenvalue), to describe the data locally. High values indicate the presence of locally linear structures. We will make extensive use of this data descriptor in Chapter 8.

7.6 Using data descriptors to evaluate embeddings

Having defined data descriptors to quantify salient properties of an embedding, how can we use them to evaluate embeddings? We first observe that within the framework of persistent homology, the data descriptors may be used as the weights in a filtration of the data. Provided we have an approximation of the connectivity in the form of a Rips graph or a Vietoris–Rips complex, this construction permits us to obtain persistence diagrams—one diagram for the original data, the others for its embeddings. We may then use the Wasserstein distance to compare persistence diagrams among each other. Figure 7.14 depicts a high-level illustration of our proposed workflow. The subsequent sections explain the individual steps in more detail.

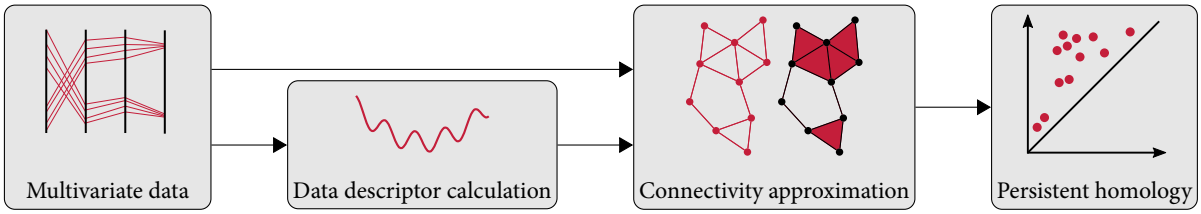


Figure 7.14: A generic data descriptor workflow using persistent homology. The result of this workflow is a persistence diagram (or a set of them) serving as a fingerprint of the data.

Throughout the remainder of this chapter, we assume that we are given a high-dimensional data set $\mathbb{X}$ and a set of embeddings, $\mathbb{Y}_1, \mathbb{Y}_2, \ldots$ generated by dimensionality reduction algorithms such as PCA or MDS.

Topological approximation

We approximate the connectivity of the original data set $\mathbb{X}$ using a Rips graph $\mathcal{R}_\epsilon$, for instance. To choose the parameters of this approximation automatically, we may use one of the heuristics presented in Chapter 5, Section 5.4, p. 96 ff.

The Rips graph $\mathcal{R}_\epsilon$ may also be expanded to a Vietoris–Rips complex $\mathcal{V}_\epsilon$. Our experiments indicate that the 0-dimensional connectivity information provided by $\mathcal{R}_\epsilon$ is often sufficiently expressive for the purpose of comparing different embeddings, though.

Data descriptor calculation

Next, we select a data descriptor—such as *density*—and calculate its values on all embeddings $\mathbb{Y}_1, \mathbb{Y}_2, \ldots$ as well as on the original data $\mathbb{X}$ Hence, we obtain a set of scalar values for each input data set. In order to account for scaling effects in different embeddings, we perform *statistical standardization* and *normalization*. More precisely, we replace each descriptor value x by

$$x' = \frac{x - \mu}{\sigma}, \tag{7.16}$$

where μ and σ are the sample mean and the sample standard deviation of the descriptor values. The tacit assumption underlying this procedure is that the values follow a normal distribution. If this is not the case, other methods from non-parametric statistics may be employed, such as the *median absolute deviation* [245].

So far, we did not make any assumptions about the properties of the data descriptor. If we want the stability results of persistent homology to be fully applicable, we require *Lipschitz continuity*, as described by Definition 4.26 on p. 73. It turns out that this is not a serious restriction. Biau et al. [46], for example, show that their density estimator is Lipschitz-

continuous. The stability theorems from Chapter 4, Section 4.6.2, p. 72 ff., thus remain fully applicable. In particular, the Wasserstein distance calculations remain stable according to Theorem 4.29 on p. 74. We will subsequently index the individual data descriptors by their embeddings. For example, we may have f_{Original}, f_{PCA}, f_{MDS}, and so on.

Persistence diagram calculation

We use the values of each data descriptor as the weights of a *sublevel set filtration*, as described in Chapter 4, p. 63, for $\mathcal{R}_\epsilon$ or $\mathcal{V}_\epsilon$. It would also be possible to calculate a new neighbourhood graph for each embedding, but this would potentially introduce instabilities into the subsequent comparisons. Keeping the domain fixed makes the assessment of topological dissimilarity more stable. Applying the persistent homology calculation from Algorithm 5 on p. 62 to the Rips graph $\mathcal{R}_\epsilon$ or the Vietoris–Rips complex $\mathcal{V}_\epsilon$ results in a set of persistence diagrams $\mathcal{D}_{\text{Original}}$, $\mathcal{D}_{\text{PCA}}$, $\mathcal{D}_{\text{MDS}}$, and so on.

Measuring global quality

At this point, we have persistence diagrams that represent the geometrical–topological properties of the selected data descriptor, both on the original data and on all embeddings. The main observation is that *any* topological distance—such as the Wasserstein distance—now permits us to quantify how well the properties of a data descriptor are preserved in an embedding. For example, assuming that we measure the density of the data and of an embedding, a small topological distance indicates that the structural features of the density function have been retained. This means that the density function in the embedding 'behaves' just like the density function in the original space. Hence, the embedding does not suffer from too many density distortions.

The persistence diagrams are well-suited for this task because they automatically encode the scale of a feature. In order to increase the topological distance by a large amount, an embedding must either not preserve *many* small-scale features or *some* large-scale features. Formally, we calculate

$$\kappa_{\text{Embedding}} := \mathrm{W}_2\big(\mathcal{D}_{\text{Original}}, \mathcal{D}_{\text{Embedding}}\big) \tag{7.17}$$

to characterize the global quality of a given embedding. This information permits us to automatically rank and sort embeddings according to their topological behaviour. The global quality $\kappa_{\text{Embedding}}$ has a natural lower bound of 0. This value would imply that the original persistence diagram and the persistence diagram on the embedding are identical, or at least indistinguishable by persistent homology. We never observed this in practice.

In the subsequent discussions, we visualize $\kappa_{\text{Embedding}}$ by arranging all embeddings along a line. The left boundary of the line corresponds to a distance of 0, meaning that the topology of the embedding coincides perfectly with the topology of the original data. The further an embedding is placed to the right, the lower its global quality value $\kappa_{\text{Embedding}}$ is. We refer to this simple visualization as the *global quality diagram*.

An upper bound for the global quality

The Wasserstein distance as used in Equation 7.17 is theoretically unbounded from above. Practically, we can use a worst-case estimate to obtain a useful upper bound. To this end, we recall the definition of W_2. Assuming that no features of the data descriptor on the original data are being preserved by an embedding, all of the points in its persistence diagram $\mathcal{D}_{\text{Original}}$ will be matched to their orthogonal projections onto the diagonal. In this case, W_2 degenerates to the sum of distances of points to their projections:

$$\begin{aligned} W_2\big(\mathcal{D}_{\text{Original}}, \mathcal{D}_{\text{Embedding}}\big) &\approx \left(\sum_{(c,d)\in\mathcal{D}_{\text{Original}}} 2^{-2}|d-c|^2 \right)^{\frac{1}{2}} \\ &= \frac{1}{2}\left(\sum_{(c,d)\in\mathcal{D}_{\text{Original}}} |d-c|^2 \right)^{\frac{1}{2}}, \end{aligned} \tag{7.18}$$

We use an approximation sign in the previous equation because we do not account for the points in the second persistence diagram, i.e. the persistence diagram of the embedding. Interestingly, this quantity is half the 2-norm of the persistence diagram $\mathcal{D}_{\text{Original}}$, a quantity introduced by Chen [93] in the context of analysing diffusion processes.

We will indicate this upper bound in the global quality diagram by a dotted line. Furthermore, all embeddings whose $\kappa_{\text{Embedding}}$ value is larger than the upper bound will be shown in grey. This colour implies that the embedding is considered unsuitable for the purpose of preserving the topology of the data descriptor.

Measuring local quality

The global quality analysis gives us information about the general suitability of an embedding. In addition, we are interested in pinpointing the regions in which an embedding fails to preserve many geometrical–topological features of the data descriptor. To this end, we propagate information from the persistence diagram distance calculations to individual points. The basic idea involves using the cascade information—i.e. the representation or

realization of a topological feature—that is associated with a point $x = (c, d)$ in a persistence diagram by Algorithm 5 on p. 62.

For dimension 0, the representation of x is a connected component in the sublevel sets of the data descriptor. If x is matched with another point y during the Wasserstein distance calculations, we extract the subgraph from the connectivity data structure—either $\mathcal{R}_\epsilon$ or $\mathcal{V}_\epsilon$—whose edges have a weight less than d. We then obtain the connected component created by x by taking a look at the simplex that created the feature x. This information may be obtained along with the calculation of persistent homology. The extracted connected component is a subset $V' \subseteq V$ of the vertices of our connectivity data structure. We assign each vertex $v' \in V'$ the cost of matching x and y while keeping track of multiple cost assignments using a list. At the end, since every vertex occurs in at least one connected component, we have at least one cost value per vertex. If multiple values exist, we use their sample mean as an indicator of the local errors that are accumulated at this vertex.

For higher-dimensional features, we use the same procedure as above but apply it to their corresponding cascades. If a high-dimensional simplex σ creates a feature, we propagate the matching costs to all its vertices $v \in \operatorname{vert} \sigma$, as well as to all the vertices of the simplices in its cascade. The matching costs accumulated by this procedure are stored in addition to the matching costs calculated in dimension 0.

Since every vertex participates in at least one connected component, we now have a local quality value for each point in the original data set. We may show these values in an embedding in order to illustrate its local quality. Following the local agreement visualization we introduced earlier, we use blue ● to indicate small errors, yellow ● to indicate medium errors, and red ● to indicate large errors. Each colour comprises one third of the range of values.

7.7 An example

As a motivating example, we use our workflow to analyse multiple embeddings of the *Swiss roll* and the *Swiss hole* data sets. The Swiss roll data set was introduced by Tenenbaum et al. [359] to show that the non-linear Isomap algorithm is capable of 'unrolling' manifolds, while linear methods fail to do so. In the following, we will use the density data descriptor and show how the global quality of different embeddings coincides with our intuition. We will then perform a local quality analysis using the Swiss hole data set.

Both data sets are sufficiently complicated to cause problems for several dimensionality reduction algorithms. At the same time, the data sets have a well-known geometrical structure and a low intrinsic dimensionality, which makes them suitable for explaining numerous aspects of our method.

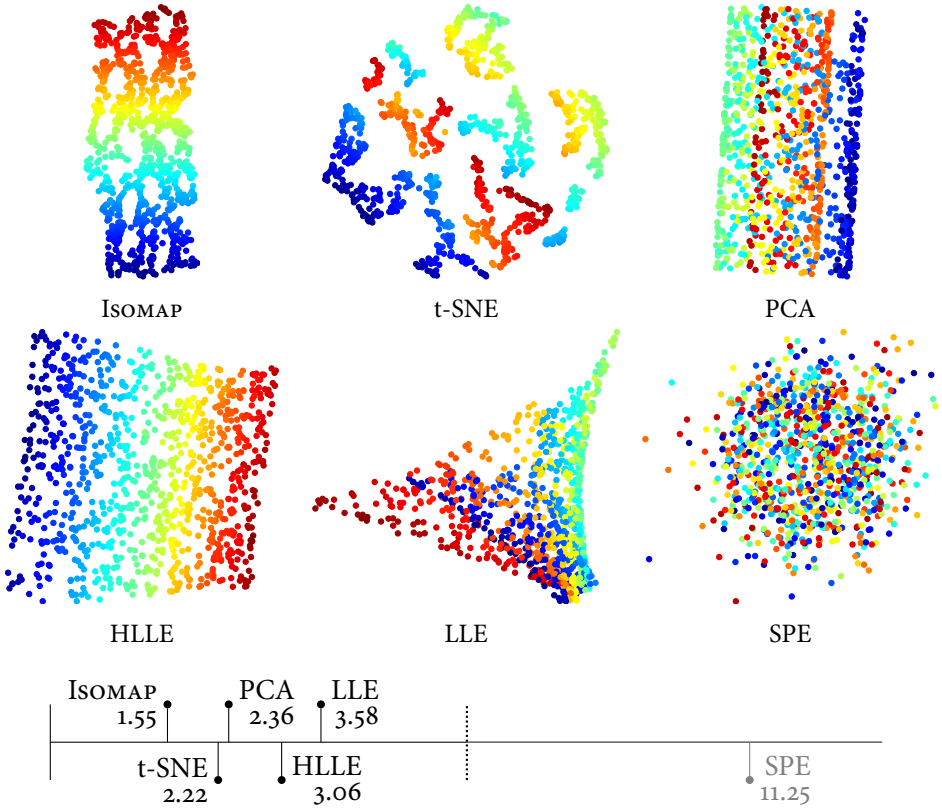


Figure 7.15: Embeddings of the Swiss roll data set, ordered by their $\kappa_{\text{Embedding}}$ values. Only Isomap is capable of 'unwrapping' the structure of the manifold properly. The dotted line indicates the worst-case bound from Equation 7.18, p. 175. Every embedding that is to the right of this line is not capable of preserving a sufficiently large amount of geometrical–topological information of the density data descriptor. We can see that this worst-case assumption is not too strict—only the SPE embedding is considered unsuitable.

7.7.1 Global quality

The global quality analysis tells us how much the density estimates in an embedding deviate from the density estimates in the original point cloud. We shall see that this corresponds to our perception of a 'good' embedding to a large extent.

Swiss roll data set

Figure 7.15 depicts multiple embeddings of the Swiss roll data set. We use the rainbow colour map—although being frowned upon when used for scientific visualization [50], it simplifies recognizing a suitable embedding. If the embedding is capable of unrolling the Swiss roll data set, we should see a perfect rainbow. It turns out that only Isomap is capable of doing this.

The other dimensionality reduction methods suffer from overplotting or other distortions. t-SNE comes relatively close in providing a useful planar representation, which is achieved at the expense of global neighbourhoods. Since our data descriptor measures density, which is essentially a *local* property of data, this embedding is not penalized as much as the other embeddings. By contrast, the embedding provided by HLLE respects neighbourhoods but introduces a large amount of distortions in the density distribution of the data—the embedding appears to have a uniform density, which is not present in the original data. Finally, the embeddings provided by LLE and SPE suffer from overplotting and, in the case of SPE, complete structural loss.

In the global quality diagram, the worst-case estimate calculated by Equation 7.18 is indicated by a dotted line. We observe that this worst-case estimate is rather tame and only considers the SPE embedding to be completely unsuitable.

Swiss hole data set

Next, we analyse embeddings of the Swiss hole data set. Here, a hole has been added to the Swiss roll data, yielding a non-convex data set that is challenging to embed. Figure 7.16 shows the embeddings of multiple dimensionality reduction schemes. Again, we use the rainbow colour map in order to make a perfect embedding easily visible. We can see that among the depicted embeddings, HLLE, t-SNE, and Isomap are capable of providing suitable embeddings. PCA and SPE—the latter despite many tuning strategies—on the other hand, feature over-plotting and distortions.

If we compare our perceptual ranking (HLLE, t-SNE, Isomap, LLE, PCA, SPE) with the ranking calculated using our method, we can see that they almost coincide. HLLE is able to unwrap the data without introducing any distortions in density. t-SNE and Isomap retain the general shape of the data, but introduce distortions. Since those distortions are of a similar nature—observe the 'squeezing' that takes place around the central hole—both methods are rated approximately equal. Furthermore, the distortions occur on comparatively large scales and thus do not result in pronounced changes in density. PCA chooses an unsuitable projection direction when embedding the data set, resulting in occlusion and overplotting. Nonetheless, it preserves density to a slightly better extent than LLE, whose asymmetrical distortions result in a lower quality value. Finally, we observe that SPE fails to embed the data set. The distortions introduced in its embedding make it almost impossible to see any relevant structures. For instance, the embedding does not indicate that there is a central hole in the data.

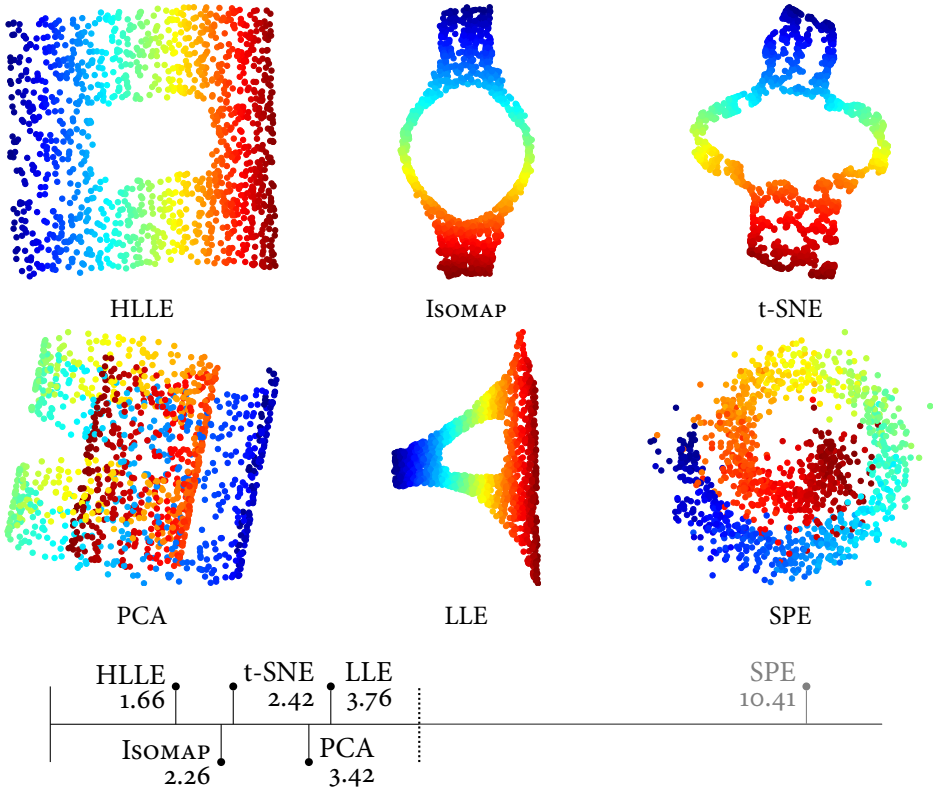


Figure 7.16: Embeddings of the Swiss hole data set, ordered by their $\kappa_{\text{Embedding}}$ values. For the most part, this order corresponds to the perceived quality of the embeddings. Note that the LLE embedding is rated slightly worse than the PCA embedding because it features a higher amount of large density errors. The dotted line indicates the worst-case bound from Equation 7.18, p. 175. Again, we can see that it is not too strict and only considers the SPE embedding to be unsuitable.

7.7.2 Local quality

We use the Swiss roll data set for investigating the local quality of some of the embeddings shown in Figure 7.15. Previously, we saw that only Isomap is capable of preserving the intrinsically planar structure of the data. The local quality measure can highlight regions in which the other embeddings—despite their inability to properly represent the data set on a global scale—preserve the density data descriptor correctly. Figure 7.17 depicts the local quality scatterplots for selected embeddings. We can see that the Isomap embedding contains almost no errors, meaning that the density is globally and locally preserved well. t-SNE, by contrast, rips the global structure of the data apart but preserves the data locally rather well. As a consequence, its local quality scatterplot is a mixture of large, medium, and small density

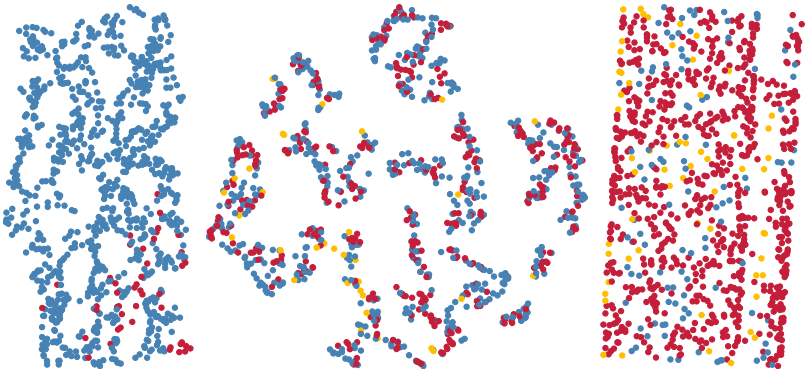


Figure 7.17: Selected local quality scatterplot for the Swiss roll. The visualization is based on a *local* assessment of errors, so it cannot indicate whether the baseline error is high or low. By construction, the Isomap embedding contains almost no density errors. We use the categorical colours introduced earlier to indicate small ●, medium ●, or large ● errors.

errors. Finally, PCA results in large distortions and folds the different layers of the data over each other. This process introduces errors on both the local as well as the global scale.

7.8 Stability & performance

The data descriptor workflow that we introduced is different from the usual filtrations that are encountered in persistent homology. Initially, we use the distances in the high-dimensional data set to obtain a Rips graph $\mathcal{R}_\epsilon$. However, we then use the values of *another* scalar function to obtain the filtration. Despite the stability results of persistent homology, this usage—which Carlsson [67] refers to as *functional persistence*—may potentially introduce instabilities into our workflow. To verify that this is not the case, we performed several robustness experiments. After reporting their results, we conclude the section with a short discussion about performance.

Stability of the ranking

The global quality ranking obtained using our workflow is stable with respect to noise in the data. To prove this, we slightly perturb the function values of the different embeddings and show that no large changes in the global ranking will occur. An alternative would be to perturb the high-dimensional data points directly and re-calculate the embeddings as well as the function values. The problem with this approach is that it is subject to instabilities that arise from the dimensionality reduction methods themselves. For example, we shall see in Section 7.9.1 that Isomap is sensitive with respect to its parameters, confirming previous objec-

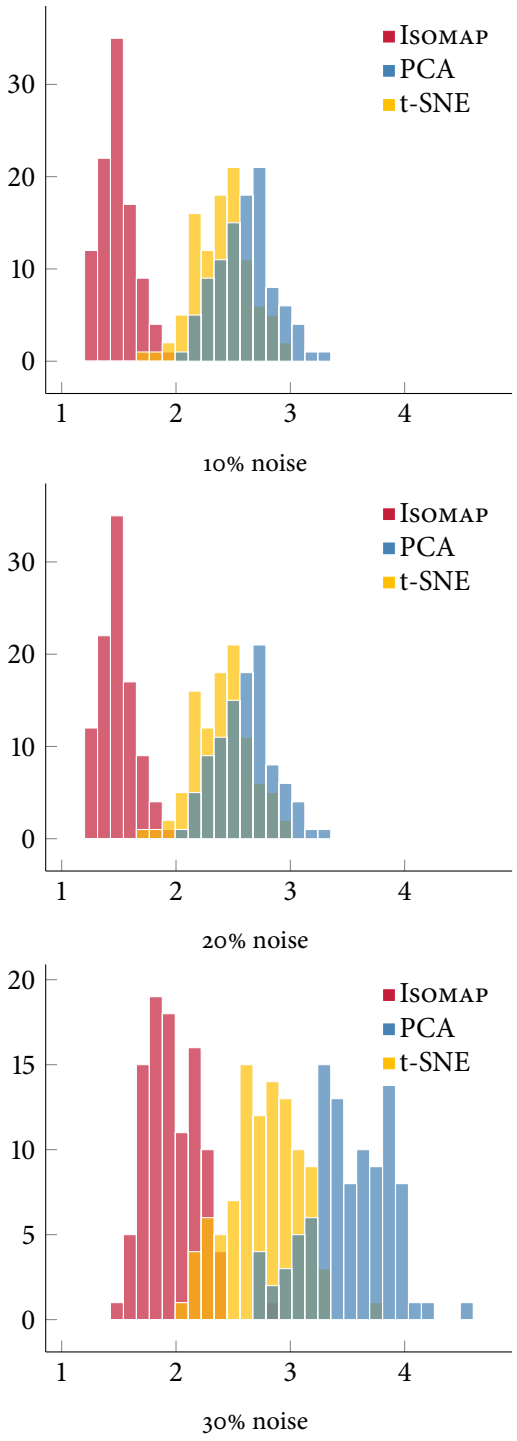
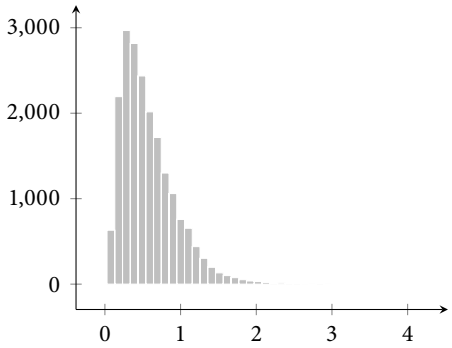


Figure 7.18: Histograms of the global quality values at different noise levels. Even when increasing the amount of noise to 30%, the values of the best-performing algorithm, Isomap, remain well-separated from the remaining dimensionality reduction algorithms. We use the same coordinate system for all histograms in order to simplify their comparison.



Minimum	0.03
Maximum	4.12
Mean	0.58
Median	0.49
Skewness	1.54
Kurtosis	7.31

Figure 7.19: Histogram of density errors on the Swiss roll data set. Errors are accumulated for $k \in [10, 20]$. The underlying distribution is the maximum absolute difference between the different density estimates for each point. Almost all errors are well within the inter-point distance of the data.

tions raised by Balasubramanian and Schwartz [21]. Consequently, we leave the embeddings as-is and only perturb the function values.

We model noise by a Gaussian distribution with $\mu = 0$ and σ as a percentage (10%, 20%, 30%) of the average difference between function values. This distribution of perturbations simulates small-scale noise, because only the local structure of the function values on the Swiss roll is distorted. Figure 7.18 shows histograms of the global quality values. We observe that the differences between Isomap—the best-performing method on the Swiss roll data—and the remaining methods remain significant for all noise levels. Even though the global quality values of the methods start to exhibit a larger variance, their relative order is always maintained. A Kolmogorov–Smirnov test [386, p. 245] shows that the quality value distribution of Isomap is statistically different from the two remaining distributions.

Stability of the parameters

Since we use a Rips graph $\mathcal{R}_\epsilon$ for approximating the connectivity of the data, we already know that it is stable with respect to small perturbations in the approximation parameter ϵ [92]. Hence, it is sufficient to verify the stability of the density estimator.

To verify the theoretical stability properties [46, 89] experimentally, we calculated density estimates for $k \in [10, 20]$ on the Swiss roll. This results in 10 different density estimates per data point. We assign each original data point the maximum difference between its density estimates and look at the corresponding distribution of values. In an ideal situation, this distribution should exhibit a large peak near zero, with a sharp decline towards higher values. Figure 7.19 shows a histogram of these errors. We observe a similar behaviour, with the majority of values being in $[0, 1]$. This interval is well below the average inter-point distance of

Method	Time
HLLE	0.23 s
Isomap	0.27 s
LLE	0.15 s
PCA	0.16 s
SPE	0.21 s
t-SNE	3.51 s
Persistent homology	0.53 s

Swiss roll

Method	Time
HLLE	0.31 s
Isomap	0.38 s
LLE	0.23 s
PCA	0.26 s
SPE	0.20 s
t-SNE	4.48 s
Persistent homology	0.46 s

Swiss hole

Table 7.3: Timing information for two test data sets. The persistent homology calculations are not a bottleneck in the analysis process. Their calculation usually requires about the same amount of time as calculating one of the embeddings does.

3.96, meaning that density estimates only vary within very small neighbourhoods and remain globally stable, even over larger ranges for k. These results agree with our previous analysis of the density estimator in Chapter 5, Section 5.4, p. 96 ff., where we analysed the densities of smaller synthetic data sets.

Performance

We performed all analyses on an Intel i7 960 machine with 8 GiB RAM. Our implementation currently uses only a single core. Since for most data sets the calculation of 0-dimensional persistent homology proves to be sufficient, the performance analysis of Section 4.3, p. 51, applies. Consequently, calculating persistent homology requires approximately linear time.

✦

Table 7.3 shows some example timing information. We can see that even though the Wasserstein distance has a worst-case complexity of $\mathcal{O}\left(m^3\right)$, where m is the number of points in the largest persistence diagram, the total time spent for our analysis is negligible because of the small sizes of the persistence diagrams involved in our analysis. The runtime is hence dominated by calculating the individual embeddings. This also holds for real-world data. However, in case the Wasserstein distance becomes computationally prohibitive, approximative graph matching algorithms [83] may be used. Recent results by Kerber et al. [220] yield more performance improvement strategies by exploiting the geometrical structure of persistence diagrams. This is done at the expense of precision, though.

7.9 Results

Having set up the workflow, explained its individual components, and proved their stability, we now take a look at how it may be used to support users in choosing a suitable dimensionality reduction method. Moreover, we shall also demonstrate how our workflow yields a better understanding of cases in which some dimensionality reduction methods fail to provide a useful embedding.

7.9.1 Synthetic faces

This data set was initially introduced by Tenenbaum et al. [359] as a showcase for non-linear dimensionality reduction methods such as Isomap. It contains 698 images with a resolution of 64 px × 64 px, each depicting a synthetic model of a human head under varying lighting and pose conditions. Figure 7.20 shows an excerpt of the data. By construction, the images are situated on an intrinsically three-dimensional manifold, parametrized by two pose variables (left–right, up–down) and one lighting variable (characterized by an azimuthal angle). A suitable dimensionality reduction algorithm should result in an embedding whose axes approximately reflect these variables or combinations of them. For layout reasons, we will refrain from showing individual faces in the resulting embeddings. We will merely comment on some of their structural features.

In the following, we shall focus in particular on different embeddings created with the Isomap algorithm (for varying neighbourhood parameters) because it exhibits interesting behaviour. Some of the embeddings are shown in Figure 7.21. The global quality visualization indicates that the Isomap embeddings are somewhat volatile with respect to k, the number of neighbours used during the neighbourhood graph construction that is central to Isomap. We reproduced the original embedding reported by Tenenbaum et al. [359] with $k = 8$, but our algorithm does not consider it to be the best embedding. In fact, there is a substantial difference between the global quality of the MDS embedding and the best Isomap embedding. The worst-case bound shows that some of the embeddings are unsuitable. This is caused by the large amount of bending that appears in the Isomap embeddings.



Figure 7.20: An excerpt of the synthetic faces data. The faces are situated on a three-dimensional manifold, which is characterized by two pose variables and one lighting variable. The absence of noise and the clear definition of the three variables make this a suitable test data set for dimensionality reduction algorithms.

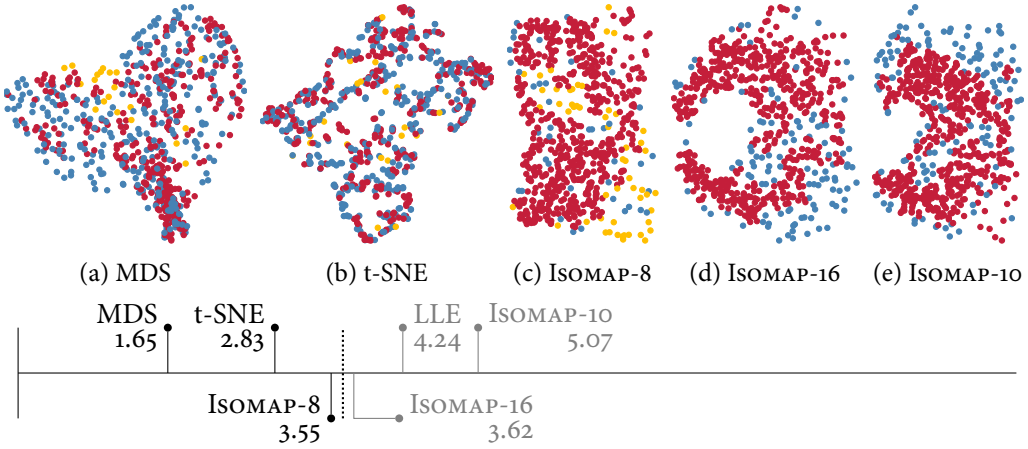


Figure 7.21: Local and global quality diagrams of the synthetic faces data. Embeddings generated using Isomap are unstable and start to bend upon increasing the neighbourhood parameter k. This leads to the interesting result that Isomap embeddings with a larger number of neighbours, such as Isomap-16, are considered to be more suitable than Isomap-10. The dotted line indicates the worst-case bound from Equation 7.18, p. 175. The visualization depicts small ●, medium ●, or large ● errors using the categorical colours introduced above.

Density errors in Isomap embeddings

By showing the local quality in a scatterplot, we observe a large amount of density errors in Figure 7.21c, Figure 7.21d, and Figure 7.21e. Paradoxically, the embeddings get worse when increasing k, but start to get better after $k = 14$. This is caused by the neighbourhood graph of Isomap. For higher values of k, this graph will approximate the connectivity of a complete graph. In this case, the Isomap embedding will degenerate to an MDS embedding. We can observe this approximation by the 'bends' that start to occur for large values of k. The two strands that are visible in the Isomap embedding for $k = 16$ will fold back onto themselves and form the long strand in the MDS embedding.

Other embeddings

t-SNE only distorts density in large neighbourhoods, but preserves it well in local neighbourhoods. Hence, its embedding—which partitions the data into smaller groups of extremely similar images—is considered more suitable than the Isomap embeddings. The embedding introduces small-scale holes that are artificial and thus get penalized by the local quality measure. Finally, the MDS embedding exhibits a fully-contrasting behaviour. Here, high density errors are predominantly confined to a single region in the lower part of the embedding. This region contains faces that are only partially lit. The Euclidean distance that we used to calculate

distances between different images loses its discriminative power for these images. The higher density of this region consequently appears to be an artefact of the embedding method. In the remaining regions, the local errors are quite low, meaning that MDS is representing the density properly here.

7.9.2 Concrete compressive strength

After analysing the synthetic faces—a data set with a clear manifold structure—we now want to analyse a data set where the internal structure is less apparent. We already encountered the *concrete compressive strength* data set in Section 7.4.2, where we performed an agreement analysis of different quality measures. While we were able to shed some doubt on the veracity of the representations of some features in an MDS embedding of the data, we now want to focus on determining which algorithms result in a suitable embedding.

In the following, we shall pay particular attention to how different algorithms represent subgroups and substructures. A previous analysis by Gerber et al. [176], based on the Morse–Smale complex, showed that the parameter space of this data set contains numerous linear substructures. In our experiments, only some dimensionality reduction algorithms display these structures in an obvious manner. Figure 7.22 shows an overview of numerous embeddings of the data. From the global quality information, we can see that t-SNE and MDS are among the best-performing methods on these data.

t-SNE and MDS embeddings

Taking a look at the t-SNE embedding, as shown in Figure 7.22a, we observe that t-SNE partitions the parameter space into smaller groups of mixtures. Global shape information gets lost by this approach. The local quality measure indicates that some parts of the parameter space exhibit a large amount of density errors. t-SNE seems to suggest an overall uniform distribution of mixtures in the parameter space, which does not always coincide with the density estimates. Thus the scatterplot contains numerous red regions. Furthermore, t-SNE does not preserve any linear structures in the parameter space.

By contrast, these structures are easily visible in the MDS embedding, as shown in Figure 7.22b. Their colour indicates that—for the most part—they are not structural artefacts of the projection. Some density misrepresentations occur, though, requiring attention in further analysis steps. Comparing this to the previous agreement analysis from Section 7.4.2, the assessment by the MRRE measure now appears to be justified, even though it was the only quality measure that considered the linear structures to be well-embedded.

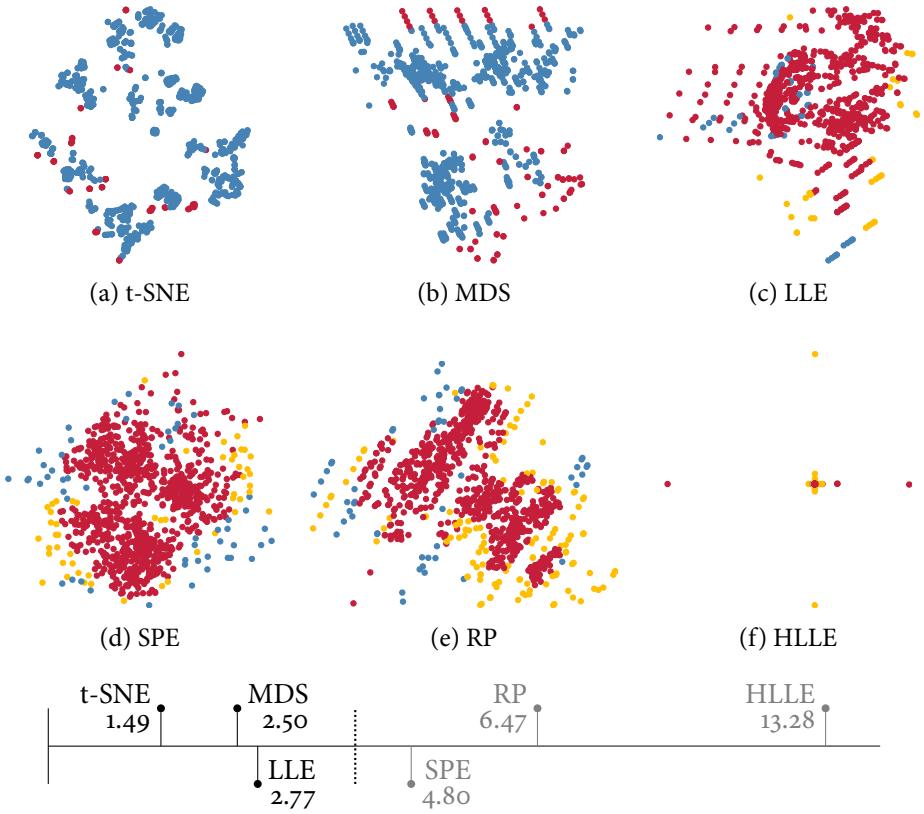


Figure 7.22: Local and global quality diagrams of the 'Concrete compressive strength' data. Linear substructures are visible in many embeddings, but they do not necessarily correspond to the same structures. In the t-SNE embedding, these structures are completely lost in favour of a grouping of very similar concrete mixtures. We observe that small ●, medium ●, and large ● errors are not distributed uniformly in the embeddings. Furthermore, embeddings in the bottom row are deemed unsuitable according to Equation 7.18, p. 175.

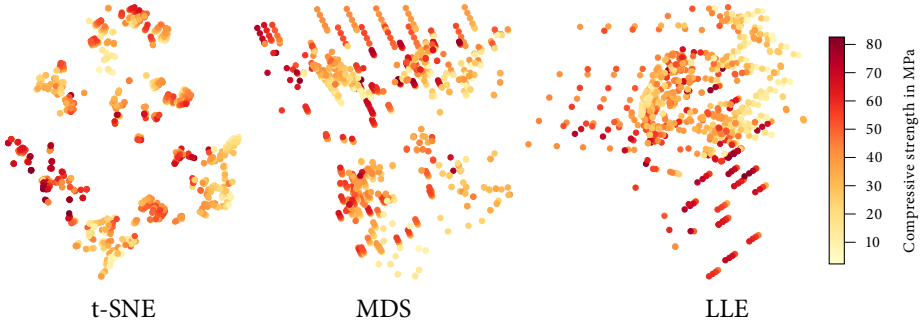


Figure 7.23: Visualizing the output variable of the 'Concrete compressive strength' data. Since all three embeddings are considered sufficiently suitable, we may use them for EDA.

Other embeddings

Several other dimensionality reduction methods did not perform well on this data set. In the RP embedding, depicted in Figure 7.22e, linear structures are visible, for example, but the density distribution becomes distorted. If we run the algorithm multiple times, the global error varies in a range of 5.88–7.89. The parts that appear dense in this embedding are hence a misrepresentation. HLLE performed even worse on this data. Despite multiple runs and a large amount of parameter tuning, the embedding does not get sufficiently better. Nothing about the structure of the compressive strength data would suggest that it cannot be embedded by this particular dimensionality reduction method. This example illustrates that caution should be exercised when applying dimensionality reduction algorithms. At the very least, multiple runs are necessary in order to confirm whether some structures are artefacts or salient features.

Using suitable embeddings for data analysis

Figure 7.23 shows the t-SNE, the MDS, and the LLE embeddings, all of which are considered suitable by our workflow because they have similarly low $\kappa_{\text{Embedding}}$ values. We use the embeddings to visualize the *output variable*—i.e. the eponymous compressive strength of a given mixture. In all embeddings, we observe that mixtures with a high compressive strength are situated next to mixtures of low compressive strength. Since the local density is represented well in most of the regions of each embedding, we may be confident that this is an actual feature of the data. We thus conclude that one issue with the compressive strength data is the *instability* of mixtures. If similar mixtures may have extremely different strengths, modelling a relationship between the different attributes becomes more complicated.

Attribute	Unit	Type
Air temperature	°C	Continuous
Surface temperature	°C	Continuous
Atmospheric pressure at sea level	bar	Continuous
Total precipitation	$\mathrm{kg\,m^{-2}}$	Continuous
Wind velocity in u-direction	$\mathrm{m\,s^{-1}}$	Continuous
Wind velocity in v-direction	$\mathrm{m\,s^{-1}}$	Continuous

Table 7.4: Attributes in the climate data set. We removed the positional information in order to focus on analysing the complete parameter space instead.

7.9.3 Climate data

Climate research is a science that relies on large-scale numerical simulations with high predictive power. This requires modelling world climate at increasingly fine resolutions, which in turn results in a large variety of complex multivariate data sets. In the following, we shall analyse a large multivariate data set from the *German Climate Computing Centre* (DKRZ) that contains simulation data of scenario A1B of the Intergovernmental Panel on Climate Change (IPCC). Said scenario represents a 'more integrated world with a balanced emphasis on all energy sources'. The data set covers a period of one year, containing a grid of 192×96 different locations on Earth and six continuous variables. Table 7.4 gives a description of the different variables and their types.

For the subsequent analysis, we will use a random sample of 1,000 points of the meteorological autumn season (September–November). This sampling was necessitated because the original data set of 18,432 data points is prohibitively large for some dimensionality reduction methods. In general, data sets from climate science are challenging because their parameter space commonly lacks well-defined clusters [213]. A suitable dimensionality reduction method should be capable of faithfully representing density in the data such that users can see whether the amount of measurements of a certain type differs, for example.

Similarities in embeddings

Figure 7.24 depicts some embeddings of the data. The global quality diagram shows that SPE, PCA, and LLE are performing similarly. Interestingly, their embeddings resemble one another. All three algorithms exhibit a dense core structure, with some separated strands, as well as several outlying points. The good performance of PCA is consistent with a hypothesis of van der Maaten et al. [257], who observed that PCA does often outperform non-linear dimensionality reduction methods on real-world data sets.

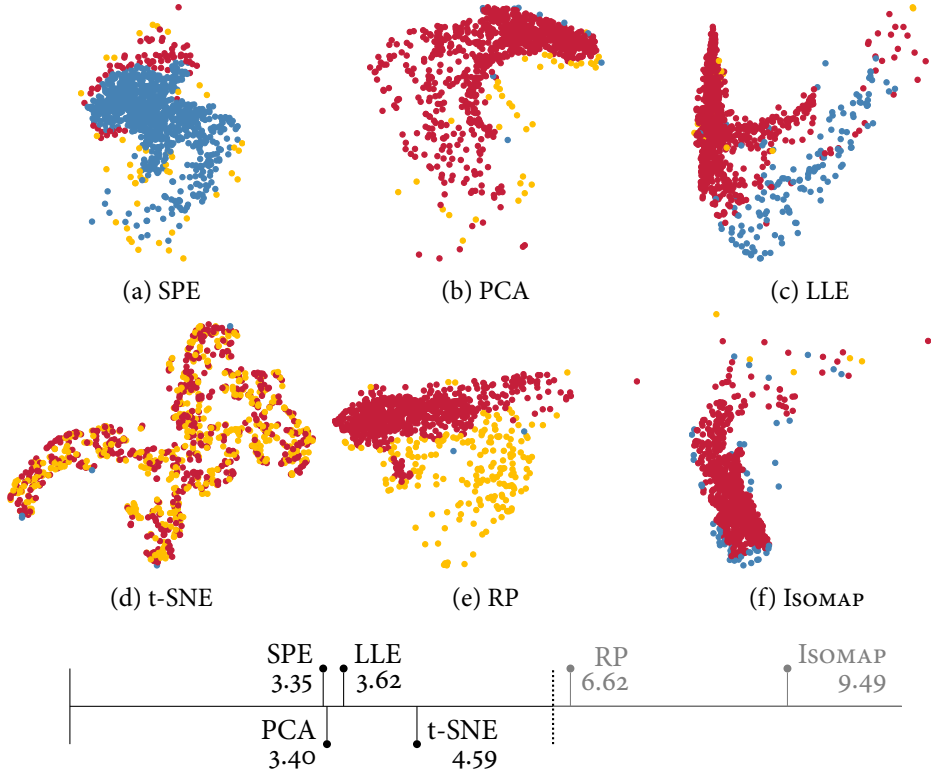


Figure 7.24: Local and global quality diagrams of the climate data. The two embeddings with the best quality, SPE and PCA, both feature branching structures and a dense core. In the PCA embedding, the branches are more pronounced. We shall subsequently see that the corresponding data points turn out to be outliers. The dotted line indicates that two embeddings are unsuitable according to Equation 7.18, p. 175, because their errors are too large.

In the local quality scatterplots, we see that all three suitable embeddings are misrepresenting the density function in the core region of the data. This is less pronounced for the SPE embedding than for the other two embeddings. Using the global quality diagram, we can see that these errors are rather small in total. As a consequence, all three embeddings appear to characterize the density of the original data quite well.

The existence of a core region of high density can be explained by the way the data set has been generated. Since every point represents a measurement from the meteorological autumn season, many measurements should coincide because they describe a similar climate. As we move away from the core of similar measurements, we tend to find only outlying measurements in a sense. The 'brushing+linking' [63, 130] paradigm reveals that these points correspond to somewhat anomalous measurements. Figure 7.25 depicts a PCP of them to show that they

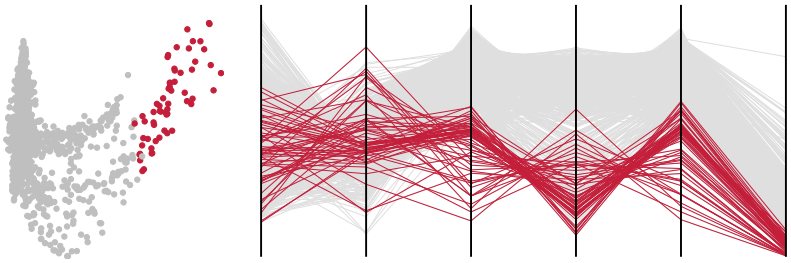


Figure 7.25: Outliers in the climate data. If we brush some of the outlying points in the LLE embedding, we can see that they correspond to anomalous measurements in the data. The PCP shows that those measurements have extremal values in several of their attributes, making them outliers in a statistical sense. For illustration purposes, we do not show labels in the PCP.

are outliers with respect to the value distributions of their individual attributes. Using these embeddings, an analyst can thus quickly decide whether two data sets—generated by different simulation runs, for example—exhibit similar homogeneity and large-scale behaviour.

Structural artefacts

The remaining embeddings suffer from structural artefacts. t-SNE, for example, attempts to group similar measurements next to each other. While this worked well for other data sets, it results in a loss of almost all density information here. Figure 7.24d shows local quality information for t-SNE; we observe that that density is misrepresented in most parts of the embedding. Consequently, anomalous measurements cannot be easily recognized here. The t-SNE embedding thus belies the fact that the data are homogeneous for the most part.

For the RP embeddings, the differences between a suitable representation of the density function and an unsuitable one can be quite subtle. The appearance of the example embedding depicted in Figure 7.24e is similar to the apperance of the LLE embedding, but its density distribution is markedly different because it suffers from overplotting. Therefore, it is considered unsuitable according to Equation 7.18, p. 175. Similar observations apply to the Isomap embedding shown in Figure 7.24f. Our new persistence-based quality measure makes it possible to detect these subtle misrepresentations and thus gives us a better justification for choosing or eschewing a certain dimensionality reduction algorithm.

7.10 Discussion

This chapter showed how novel variants of persistent homology can be used to evaluate dimensionality reduction methods. We covered two different viewpoints to connect our methods to existing research. First, after discussing existing quality measures, we showed how to compare their *agreement* on embeddings of the data in a systematic and stable manner, using

a persistence-based scalar field decomposition algorithm. The agreement of different quality measures was visualized both globally, in the form of the *relative agreement diagram*, and locally on the embedding. Our method remains applicable for high-dimensional embeddings that cannot be visualized directly. We furthermore demonstrated the capabilities of this method. In contrast to existing approaches, it permits a rapid and robust comparison of multiple quality measures. The shortcoming of our method is that it cannot be easily used to choose a suitable dimensionality reduction algorithm—we can only use it to compare the behaviour of quality measures on a given embedding, but we need to trust their assessment.

Consequently, as a second viewpoint, this chapter introduced a generic framework based on persistent homology that can be used to select appropriate dimensionality reduction algorithms. The framework employs data descriptor functions, such as density, to obtain persistence diagrams of the original data and its embeddings. We then described how to measure global quality by calculating distances between the persistence diagrams of the embeddings and a reference persistence diagram of the original data. Moreover, we provided an upper bound for these distances. We then showed how to employ local quality scatterplots to highlight regions of low or high quality to users. This permits users to quantify whether an embedding is faithful to features in the original data.

We demonstrated the stability of the framework with respect to random perturbations of the input data. We then analysed multiple data sets of varying complexities and showed how to use both global and local quality information to judge the quality of embeddings. Surprisingly, we found that PCA and MDS sometimes manage to outperform more complex non-linear dimensionality reduction methods.

Agreement analysis: Extensions & future work

We used a persistence-based scalar field decomposition algorithm to calculate the agreement between different quality measures. An interesting point for future work thus concerns the systematic evaluation and comparison of other scalar field partitioning algorithms. In particular, the partitioning scheme could be compared to previous work by Schneider et al. [325, 326], who used *merge trees* for decomposing a scalar field.

Another facet of our agreement analysis concerns investigating different similarity measures. Here, we used the *Jaccard distance* [402, p. 435], but there are many competing similarity measures, such as measures based on *mutual information* and other information-theoretical measures. In comparison with the simple Jaccard distance, these measures could conceivably improve the results presented here. Of particular interest could be the measures by van Dongen [373] and Meilă [264] because of their scaling and robustness properties.

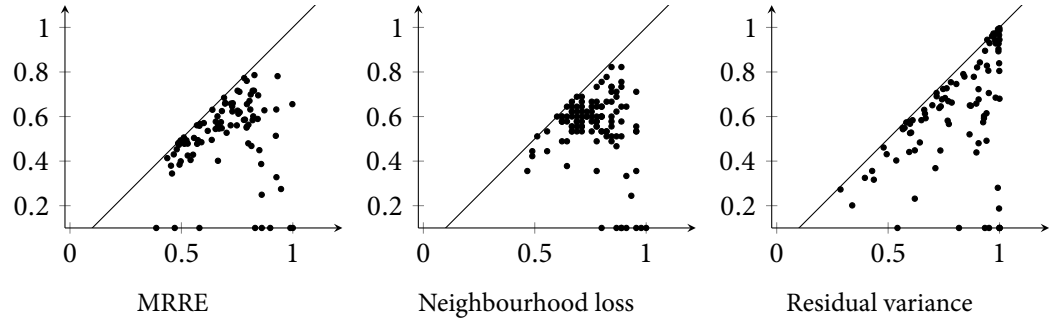


Figure 7.26: Selected persistence diagrams with good separation properties. All persistence diagrams for the t-SNE embedding of the handwritten digits data exhibit good separation properties, which simplifies their decomposition.

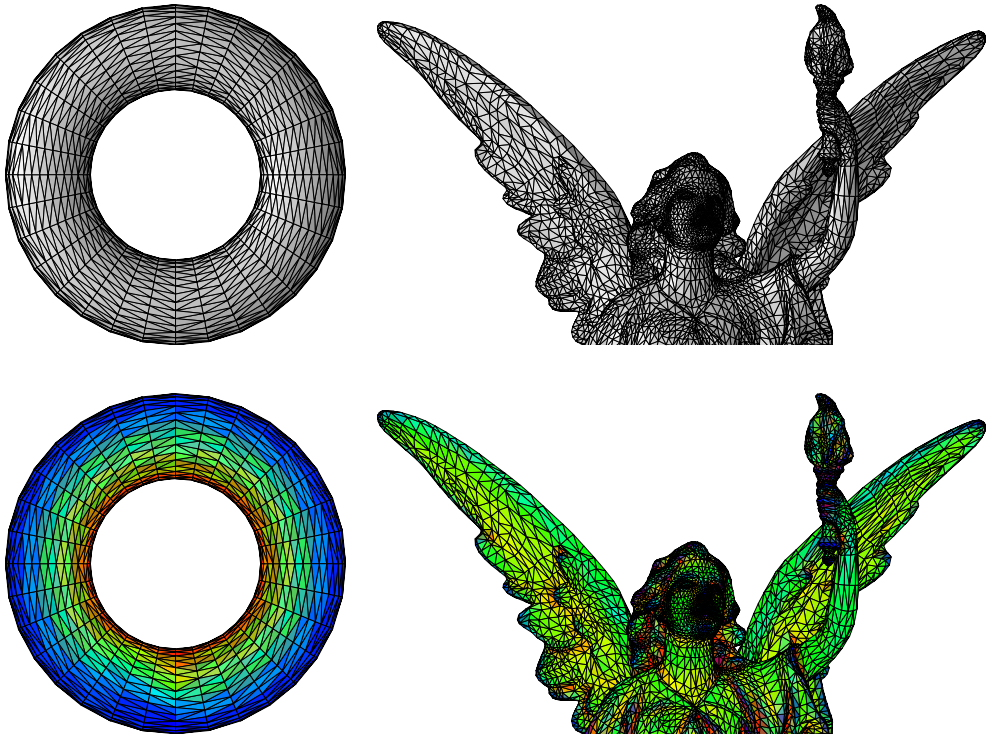
In addition, the analysis of separation criteria for persistence diagrams might prove very fruitful. The persistence diagrams for the data we analysed in this chapter exhibit very good separation properties; see Figure 7.26, for example. For the agreement analysis, we have described a way of determining whether a persistence diagram may be separated properly. However, in cases where our algorithm fails to do so, we currently do not know whether there really is no separable structure present or whether the extraction of topological structures could be improved by further processing. In this case, an algorithm by Kloke [222] could be used to de-noise the data in order to 'strengthen the topological signal'.

Data descriptor analysis: Extensions & future work

Since the data descriptor framework is not restricted to visualizations, it could also be used in a more generic data mining context for the purposes of *feature selection* or *intrinsic dimensionality estimation* [374]. By successively removing attributes from the data until the quality falls below a certain threshold, we may obtain compressed versions of a data set in which the most important geometrical–topological features are preserved.

A natural extension of this framework would be the integration of different neighbourhood graphs proposed by Correa and Lindstrom [115]. Not only could this further improve the stability, it would also permit the integration of the framework into a more traditional Morse–Smale complex analysis process. The framework would also benefit from the investigation of different distance measures, especially those that are based on high-dimensional features, such as the one proposed by Lee et al. [237], or the usage of *metric learning* [122, 228].

Furthermore, we do not claim that the data descriptors presented in this chapter are already the most salient ones. Future work should thus foremost centre on analysing the suitability of different functions for topological data analysis. Especially in the context of computer graphics or data mining, different descriptors or different notions of shape are employed



Torus Lucy

Figure 7.27: An illustration of curvature. Both models are coloured using their *mean curvature* [236, p. 157] and a standard rainbow colour map. Red values indicate regions of high curvature.

and our framework can benefit from them. The *Laplace–Beltrami operator* [385, pp. 220–222] on a Riemannian manifold is an ideal candidate. It has already proved to be useful for shape segmentation in low-dimensional data such as meshes [307, 308]. Furthermore, the *heat kernel signature* has been derived from the Laplace–Beltrami operator [352], resulting in provably stable and informative descriptors of the shape of an object in three dimensions. This signature may be modified to be scale-invariant [57], but there are still no good heuristics for choosing suitable parameters for the underlying diffusion process. A generalization to higher-dimensional data—in particular point clouds—turns out to be rather challenging, though. An approach of Belkin et al. [33] requires a Delaunay triangulation, which has a complexity of $\mathcal{O}\left(n^d\right)$ for arbitrary dimensions d.

Moreover, many shape descriptors in 2D and 3D already make use of concepts from differential topology such as *curvature*, as Figure 7.27 illustrates—it would only be natural to extend this to higher dimensions. Collins et al. [109] and subsequently Carlsson et al. [72] already showed that low-dimensional curvature information can be used to obtain shape descriptors based on persistent homology. Unfortunately, these ideas were not pursued further.

One reason for this may well be the complexity of calculating or estimating curvature in higher dimensions—instead of dealing with principal curvatures as in lower dimensions, high-dimensional manifolds require calculating the *Ricci curvature tensor*, for instance. Nonetheless, the author considers it possible to extend *integral invariants* [299] in order to approximate *mean curvature* [236, p. 157] in higher dimensions.

On the theoretical side, the stability of persistent homology with filtrations based on arbitrary function values, which Carlsson [67] refers to as *functional persistence*, should be investigated more thoroughly. Our experiments in Section 7.8 as well as our results indicate that the approach is indeed stable with respect to noise in the function values, but so far, there are no formal stability results. The data descriptor analysis shows that such filtrations can be very beneficial for quantifying structural differences of functions defined on multivariate data sets.

8 Landscape metaphors for multivariate data

A common issue when dealing with multivariate data sets is the quantification of their differences and similarities. In the previous chapters, we visualized topological properties—both directly and indirectly—in order to obtain information about salient features. Furthermore, we quantified the distances of certain data sets to a reference data set. In this chapter, we work under the assumption that no reference data set exists but we nonetheless need to assess a large amount of different multivariate data sets by some means.

✦

A very powerful metaphor that directly exploits human perception is the usage of *landscapes* for the purpose of information visualization. Points—or glyphs in general—are arranged via some similarity measure. The resulting grouping already yields a lot of interesting information. We will examine the landscape metaphor under the lens of topological methods in this chapter. Moreover, we will exemplify how a novel workflow based on topological distances helps uncover complex relationships in data.

In the first part of this chapter, we will visualize structural differences of various regression analysis models. Existing approaches ignore the structural information of the model. We will see, however, that including this information helps increase the expressive power of the analysis. In the second part of this chapter, we will analyse a collection of embeddings of high-dimensional data sets under multiple aspects. To the best of our knowledge, no comparable technique for providing such an analysis exists. This chapter is based on two publications [311, 313] by the author.

8.1 Visualizing regression analysis models

Mathematical models are commonly used to describe phenomena in different scientific disciplines. Whether it is chemical processes in the life sciences, numerical simulations in automotive engineering, or voting simulations in the social sciences: The goal of a model is to provide an adjustable—of necessity simplified—representation that permits and facilitates

a detailed analysis of a particular phenomenon. Regardless of their concrete mathematical representation, all models are conceptually the same. Given a set of input variables, such as a feature-based description of a molecule, a model predicts one or more output variables, such as a set of chemical properties.

Since models are unable to capture all aspects of a system with infinite precision, each model is prone to introduce noise and errors to an analysis. This issue is often mitigated by developing different models concurrently and letting them compete on a set of test data. Given finite resources and finite time, it is not straightforward to choose a good model, though. Different approaches usually differ in their computational complexity, their accuracy, and many more parameters. Hence, the quantification of differences between the models is an integral task in these workflows. Differences in predictive power can be easily measured, provided test data is available. *Structural differences*, on the other hand, cannot be quantified as easily, but play a decisive role in the performance of a model [227, p. 61 ff.]. They appear to have been ignored in the literature so far. The basic idea is that a model should be true to the structure of the original data in order to yield a usable description. Subsequently, we will develop a new workflow and a new visualization, the *model landscape*. It enables scientists to assess the suitability of competing models in a quantitative and qualitative manner.

8.1.1 Related work

A common approach to measure structural differences is to perform an analysis of the parameter space of a model. This type of model analysis has already been the subject of previous research. In recent work, Sedlmair et al. [330] developed a conceptual framework for describing numerous approaches to parameter space visualization. Among the earliest such methods is the one by Spence et al. [348], who introduced the *prosection matrix* to visualize the influence of certain parameters in a functional design process. Exploring the parameter space is further facilitated by scatterplots and scatterplot matrices of a predefined set of parameter values, such as design tolerance specifications. Similarly, Bruckner and Möller [59] developed an exploration process for the parameter spaces occurring in visual effects design. Their method is based on simulating the visual effect, such as a flame, with sampled sets of parameter vectors. The resulting visual effects are then clustered according to their similarity. Visualizing clusters helps artists determine which parameter values result in effects with similar looks. Bergner et al. [39] use a partitioning approach to obtain regions of distinct behaviour in the parameter space. These regions enable domain experts to learn qualitative differences in model outputs. In case the parameter space is already completely enumerated (or is amenable to this sort of analysis), a system of Matković et al. [262] permits rapid visual prototyping using direct steering of the parameter values.

So far, methods focused on a parameter space of a simulation directly. We want to approach the problem of analysing different models of certain phenomena within data. To this end, Unger et al. [372] developed a validation concept for the goodness-of-fit of models in the context of geoscientific simulation models. Mühlbacher and Piringer [274], focusing on evaluating regression models, partition input data into disjoint regions. Their approach is especially useful when analysts need to discover which variables to include in a model. Finally, Rheingans and desJardins [309] visualize the predictive qualities of different models, using both visualizations of probability distributions as well as self-organizing maps. Their approach is geared towards problems of class prediction, though.



The method presented in this chapter differs in two aspects from the previous work. First, it goes beyond the analysis or visualization of a single parameter space. Instead, it captures the predictive capabilities of a set of models with potentially very different parameter spaces. Second, it is not restricted to labelled data, special subtypes of models, or enumerated parameter spaces. Our approach solely requires the existence of a suitable dissimilarity measure on the data, such as the Euclidean distance, and the possibility to rewrite a model in terms of a scalar function $f: \mathbb{D} \subseteq \mathbb{R}^n \to \mathbb{R}$ from the data domain $\mathbb{D}$ to the set of real numbers $\mathbb{R}$.

8.1.2 Quality measures for regression analysis

In the following, we narrow our focus to the problem of *regression analysis*. Given a set of n measurements with d attributes, each instance may be represented as a d-dimensional vector x, which is usually taken to be real-valued for convenience reasons, i.e. $x \in \mathbb{R}^d$. We now assume that each measurement also has an associated scalar property, denoted $s \in \mathbb{R}$. The goal of regression analysis is to derive a functional relationship between each vector x and its associated scalar property s. There are numerous ways of performing regression analysis. In the machine learning community, for example, *support vector machines* (SVMs) [136] are often used. Rousseeuw and Leroy [321] give a detailed introduction to regression analysis in the context of mathematical statistics.

Regardless of the method used to perform regression, each results in a set of predicted values, which we will refer to as the *model* of the scalar function. We are not interested in the inner workings of an algorithm and treat it as a black box. It is common practice to partition the data into a larger *training data set* and a smaller *test data set*. Algorithms are then applied to the former for the purpose of parameter tuning. Their performance is then evaluated on the latter, usually by calculating several statistics. We shall briefly look at two state-of-the-art quality measures and outline their properties.

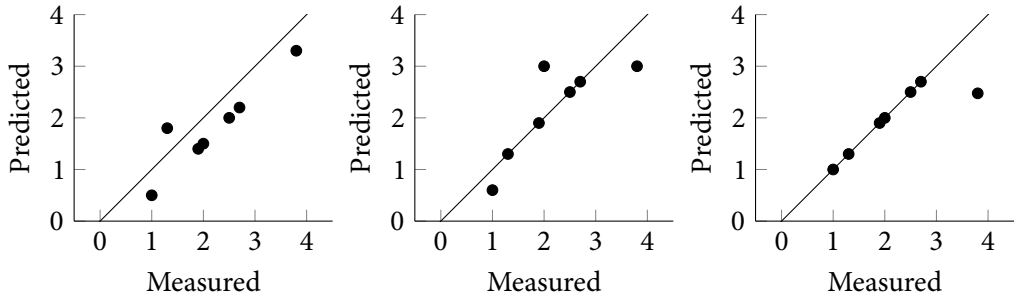


Figure 8.1: An example of the insensitivity of the RMSE. The scatterplots depict three different models with the same RMSE of 0.5. The leftmost model suffers from systematic errors. It cannot be separated from the remaining models by its RMSE alone, though.

Definition 8.1 (root-mean-square error). We already encountered the RMSE as a quality measure for dimensionality reduction methods. Here, we rephrase it for the purpose of assessing a regression model. Given a model with n predicted values $m = (m_1, \ldots, m_n)$ and n original values $s = (s_1, \ldots, s_n)$, with $m_i \in \mathbb{R}$ and $s_i \in \mathbb{R}$, their RMSE is defined as:

$$\mathrm{RMSE}(m, s) := \sqrt{\frac{\sum_{i=1}^{n} (m_i - s_i)^2}{n}} \tag{8.1}$$

In essence, RMSE aggregates squared errors in the predicted values, which means that it considers large-scale errors to be more important than small-scale errors. RMSE is not particularly sensitive because its aggregation and mean calculation is capable of masking errors in the model. Figure 8.1 demonstrates this issue by depicting three models with equal RMSE values but different behaviour.

Definition 8.2 (Correlation coefficient). The *correlation coefficient* R^2 measures how well the model and the original values are correlated. It is usually defined using *Pearson's correlation coefficient*,

$$\begin{aligned} R^2(m, s) &:= \mathrm{cor}(\{m_1, \ldots, m_n\}, \{s_1, \ldots, s_n\})^2 \\ &= \left(\frac{\sum_{i=1}^{n} (m_i - \overline{m})(s_i - \overline{s})}{\sqrt{\sum_{i=1}^{n} (m_i - \overline{m})^2 \cdot \sum_{i=1}^{n} (s_i - \overline{s})^2}} \right)^2, \end{aligned} \tag{8.2}$$

where $\overline{m}$ and $\overline{s}$ refer to the sample mean of the model values and the original values. The correlation coefficient is known to be unable to detect systematic over- or underpredictions of a model [227, pp. 95–97]. Figure 8.2 shows an example of this shortcoming.

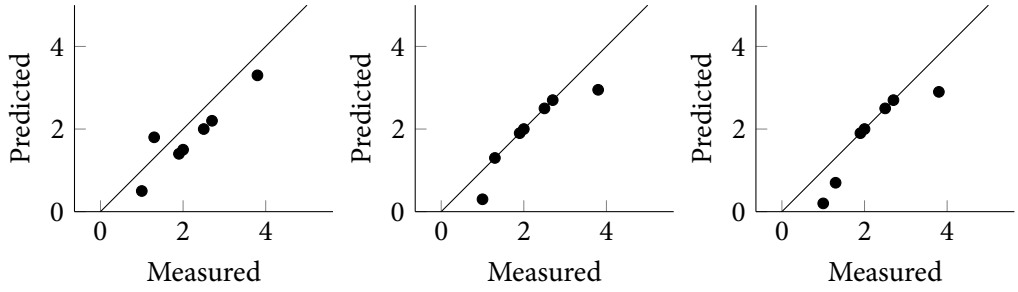


Figure 8.2: An example of the insensitivity of R^2. Again, the scatterplots depict three different models with respect to the original measurements. For each model, $R^2 \approx 0.91$. The leftmost model exhibits systematic errors for all measurements, while the two remaining models only tend to predict deviations from the original data for smaller and larger values. The correlation coefficient is incapable of distinguishing between these behaviours.

To mitigate the individual weaknesses of each measure, they are usually employed jointly. For particular applications, different measures have turned out to be more successful. The log of the accuracy ratio, which is the logarithm of the quotient of the predicted value divided by the actual value, avoids some of the issues outlined above [363], for example.

8.1.3 A quality measure based on persistent homology

We complement existing quality measures by providing one that is based on persistent homology. In the previous chapters, we have already seen that persistent homology is capable of quantifying structural changes in functions. Generally speaking, we want to quantify large-scale *structures* rather than large-scale *errors*, which are already well-captured by RMSE. As we have seen in Chapter 4, Section 4.6.3, p. 75 ff., the *Wasserstein distance* between persistence diagrams may become a suitable metric for this purpose. It turns out that the functions obtained using regression analysis can be easily integrated into our generic pipeline for persistent homology. If both the initial measurements and the predictions are scalar values, they can be used as weight functions on a simplicial complex. This leads to the following algorithm:

1. Calculate a Vietoris-Rips complex $\mathcal{V}_\epsilon$ on the original data, using any of the heuristics we have encountered in Chapter 5, Section 5.4, p. 96, ff., or a domain-specific scale parameter.

2. Use each set of predicted values and measured values as a weight function on $\mathcal{V}_\epsilon$. This results in a set of simplicial complexes.

3. Calculate persistent homology on each weighted complex, yielding a set of persistence diagrams. As in the previous chapters, each persistence diagram summarizes structural information about the data.

4. Calculate the Wasserstein distance W_2 between each pair of persistence diagrams. Collect distances of the form $W_2(\mathcal{D}_{\text{Measured}}, \mathcal{D}_{\text{Predicted}})$, i.e. distances between the original data and some model. Build a $k \times k$ matrix of the distances between different models.

5. Use *metric multidimensional scaling* [224, 225] to embed the matrix of inter-model distances into the plane. This works because the Wasserstein distance is a metric. Proximity in this embedding indicates that two models have a similar topological behaviour. We call the resulting embedding the *model landscape*. It shows the relative differences between models.

In addition to the model landscape, the distances between measured and predicted data serve as a global quality measure, similar to RMSE, R^2, and the generic method we described in Chapter 7. We may also construct model landscapes using the established quality measures. However, for this to work, we first need to evaluate to what extent existing quality measures can be used as metrics. As we shall subsequently see, they permit a reformulation to be applicable in that sense. Nonetheless, their embeddings will turn out not to be useful to indicate similarities between models—the previous discussions have already pointed out some of their shortcomings.

Existing quality measures are metrics

We first recall the properties that are required for a metric. A metric $\text{dist}(\cdot,\cdot)$ in the mathematical sense needs to satisfy the following properties:

1. Non-negativity: $\text{dist}(x, y) \geq 0$
2. Identity of indiscernibles: $\text{dist}(x, x) = 0$
3. Symmetry: $\text{dist}(x, y) = \text{dist}(y, x)$
4. Subadditivity or triangle inequality: $\text{dist}(x, y) \leq \text{dist}(x, z) + \text{dist}(y, z)$

Of the different properties, the *subadditivity* property is arguably the most important one. It states that if two objects x and y are *close* or *similar* to a third object z, they must by necessity also be close to each other. Subadditivity is thus the mathematical basis for grouping or clustering of points.

Lemma 8.3. The RMSE is a metric.

Proof. By definition, $\mathrm{RMSE}(x, y) \geq 0$. We have furthermore $\mathrm{RMSE}(x, y) = 0$ if and only if $x = y$. Likewise, the function is symmetric by construction. In order to show that the triangle equality holds, we apply a transformation:

$$\mathrm{RMSE}(x, y) = \sqrt{\frac{\sum_{i=1}^{n} (x_i - y_i)^2}{n}} \tag{8.3}$$

$$= \frac{1}{\sqrt{n}} \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{8.4}$$

$$= \frac{1}{\sqrt{n}} \|x - y\|_2 \tag{8.5}$$

In the previous equation, $\|\cdot\|_2$ denotes the usual Euclidean metric, which satisfies the triangle equality. Since the factor $1/\sqrt{n}$ is positive, the triangle equality holds for RMSE. ∎

We can obtain a similar result for R^2, provided some mild assumptions about the input data hold and we are content to use a transformed version of R^2. In this case, we have the following lemma.

Lemma 8.4. Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ be two measurements with zero mean and unit variance. This can always be achieved by pre-processing the data. Then the function

$$\widetilde{R}(x, y) = \sqrt{1 - \mathrm{cor}(x, y)} \tag{8.6}$$

is a metric. Note that this result only holds for *Pearson's correlation coefficient*.

Proof. We shall show that the Euclidean distance between x and y is a multiple of the desired function:

$$\|x - y\|_2 = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{8.7}$$

$$= \sqrt{\sum_{i=1}^{n} x_i^2 - 2 \sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} y_i^2} \tag{8.8}$$

Since x and y have zero mean, their squared sums evaluate to n. Thus, the previous equation becomes:

$$= \sqrt{2n - 2n \sum_{i=1}^{n} x_i y_i} \tag{8.9}$$

Because x and y have zero mean and unit variance, the sum is the sample correlation coefficient of x and y:

$$= \sqrt{2n - 2n \operatorname{cor}(x, y)} = \sqrt{2n}\sqrt{1 - \operatorname{cor}(x, y)} \tag{8.10}$$

$$= \sqrt{2n} \cdot \widetilde{R}(x, y) \tag{8.11}$$

As $\sqrt{2n}$ is always positive, the transformed function is a metric. ■

✦

Of course, it would be more satisfying to use the squared correlation coefficient as a metric. Since a construction similar to the proof above requires using squared Euclidean distances, however, this will not lead to a metric—the square of the Euclidean distance is not a metric in the mathematical sense because the triangle inequality does not hold. This can be seen by a simple counterexample. Let $x = (1, 0)$, $y = (0, 1)$, and $z = (0, 0)$. Then we have $\|x - y\|_2^2 = 4 \not\leq \|x - z\|_2^2 + \|y - z\|_2^2 = 2$. When calculating the model landscapes, we shall thus use the transformed correlation measure. Just like the RMSE distance, it will turn out to be rather insensitive to differences between models, though.

8.1.4 Solubility analysis

In the following, we will apply the landscape metaphor to example data from drug development. An important task in this field is the prediction of the solubility of a chemical compound. This property measures how easily the compound dissolves in a particular solvent such as water. Solubility is thus of paramount importance for any substance that is to be administered as a drug, e.g. orally or by injection.

Predicting solubility requires defining a set of descriptors of a chemical compound. These descriptors can range from a simple binary descriptor that indicates the presence or absence of a certain bond to very complex descriptors that take molecular connectivity into account. Given these descriptors and a set of compounds with known solubility values, different models may then be trained to predict solubility when being presented with a certain descriptor. The descriptors that we use in this section are part of a database of chemical compounds. Originally, they have been developed by Tetko et al. [360] for the purpose of aqueous solubility estimation. We will use a set of different models to predict solubility values. Table 8.1 lists their abbreviations and gives a short description.

Performance	Model	Brief description
High	`cubist`	Cubist regression trees
	`random forest`	Random forests
	`regression trees`	Boosted regression trees
	`svm`	Support vector machines
Medium	`lm`	Linear regression
	`m5`	Model trees
	`mars`	Multivariate adaptive regression splines
	`pls`	Partial least squares
	`ridge`	Ridge regression with penalties
	`rlm`	Robust linear regression
	`rlm pca`	Robust linear regression with pre-processing
Low	`bagged trees`	Bagged model trees
	`enet`	Regularized regression with penalties
	`knn`	k-nearest neighbours
	`rpart`	Single regression trees

Table 8.1: Models used for solubility prediction. The performance refers to how well the model is able to predict solubility after training. It was determined by Kuhn and Jonhson [227, pp. 221–223], using a combination of RMSE, the correlation coefficient, and 10-fold cross validation.

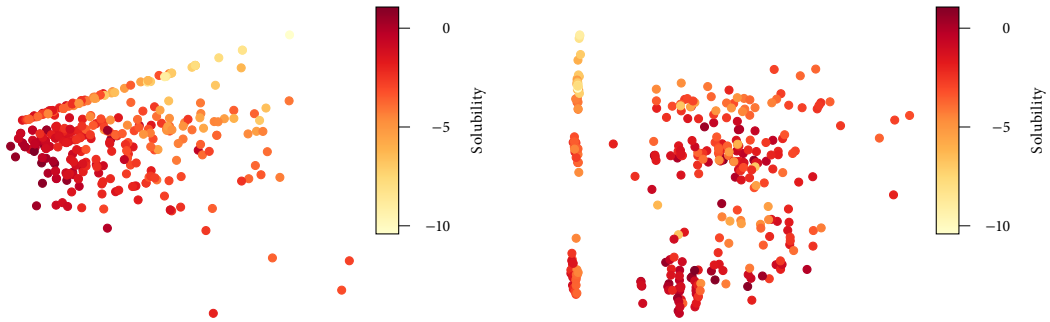


Figure 8.3: Raw and transformed molecular descriptor data. Higher solubility values are desirable. The large amount of overplotting and the lack of clear structures with respect to the solubility indicate that the intrinsic dimensionality of the data is much higher than two. Low-dimensional embeddings are thus incapable of supporting an analysis of models. Similarly, the large amount of attributes prevents visualizing molecular descriptors with standard techniques. This necessitates the use of topological techniques.

Input data

The complete set of input data is an unstructured point cloud of 1,267 compounds, each described by a 228-dimensional feature vector. The first 208 dimensions of the feature vector are binary 'fingerprints', indicating whether a certain chemical structure is present. This is followed by 16 counters, such as the number of atoms, the number of double bonds, the number of rings, etc., and 4 continuous attributes, namely the molecular weight, the hydrophilic factor, and two types of surface area measurements. Finally, each compound has an associated solubility value that is given as a dimensionless logarithm of the solubility measured in $\mathrm{mol\,l^{-1}}$. Due to the heterogeneous structure of the data, we only use the binary attributes of the molecular descriptors for the subsequent calculation of persistent homology. This enables us to use the *Hamming distance* as described by Definition 6.7, p. 133.

Pre-processing and cleaning We applied data cleaning and model cleaning as explained by Kuhn and Johnson [227, pp. 102–105]. In particular, this involves removing the skewness from the continuous descriptors by means of a Box–Cox transformation [52]. For the models themselves, we used cross-validation to perform parameter tuning on the training data. We add the label 'tuned' to a model in order to indicate this. However, the effects of parameter tuning can be neglected in most cases—only `enet` turns out to move from the group of low-performance models to the top of the medium-performance models. The remaining models only slightly change their rank when undergoing parameter tuning.

Figure 8.3 on p. 205 shows embeddings of the raw and transformed descriptor data that have been obtained using *multidimensional scaling* (MDS) [49]. It is readily visible that two dimensions are insufficient to describe the complex relations inherent to the data. This behaviour is not restricted to any particular dimensionality reduction technique. In the following, we will analyse the structure of the test data set. It contains 316 of the 1,267 descriptors. The models were trained on the remaining 951 molecular compounds.

Persistent homology

We set $\epsilon = 49$, as suggested by the scale heuristic from Chapter 5, Section 5.4, p. 96 ff., and compute a three-dimensional Vietoris—Rips complex on the data. While higher-dimensional connectivity may reveal additional information in general, here we observed that the differences in the topological approximations are minuscule at best. Following the explanation from Section 8.1.3, we apply the solubility values of each model as weights for the 0-simplices

Performance	Model	RMSE	R^2	W_2
High	`cubist tuned`	**0.60**	**0.92**	2.62
	`random forest`	0.65	0.90	2.48
	`regression trees`	0.62	0.91	2.47
	`svm`	0.64	0.91	2.81
	`svm tuned`	0.61	0.91	**2.42**
Medium	`lm`	0.76	0.87	2.91
	`lm tuned`	0.75	0.87	3.16
	`m5`	0.78	0.86	3.02
	`mars`	0.72	0.88	2.99
	`pls`	0.74	0.87	3.22
	`pls tuned`	0.74	0.88	2.83
	`ridge`	0.75	0.87	2.96
	`ridge tuned`	0.72	0.88	2.96
	`rlm`	0.75	0.87	3.85
	`rlm pca`	0.79	0.86	3.60
Low	`bagged trees`	0.84	0.84	3.24
	`enet`	**1.14**	0.80	4.77
	`enet tuned`	0.71	0.88	3.10
	`knn tuned`	1.06	**0.74**	2.97
	`rpart`	0.92	0.80	**5.16**

Table 8.2: Global quality values of all models. The largest and smallest value in each column are shown in a bold font. RMSE and R^2 agree in their assessment of the best-performing model. The Wasserstein distance W_2 is even capable of detecting salient differences between models with parameter tuning, as shown for `pls` and `pls tuned`.

during the expansion process, resulting in a set of different simplicial complexes for each model and the ground truth data.

In contrast to the calculation of individual models, which may take several minutes on an Intel i7 960 machine, the calculation of persistent homology only imposes a light burden on the complete data analysis workflow. Using a single-core implementation, the one-time Vietoris–Rips expansion takes 41.39 s. Calculating persistent homology then takes an additional 21.10 s per model, followed by a mere 0.67 s for the calculation of distances and the embedding.

Comparison with original data

We first compare each model to the original data. This yields a ranking by quality values. Table 8.2 shows a comparison between RMSE, R^2, and the Wasserstein distance W_2 for all models. We can see that the three measures agree at least in their assessment of the different groups—the best value will always be achieved by a high-performance model, while the worst

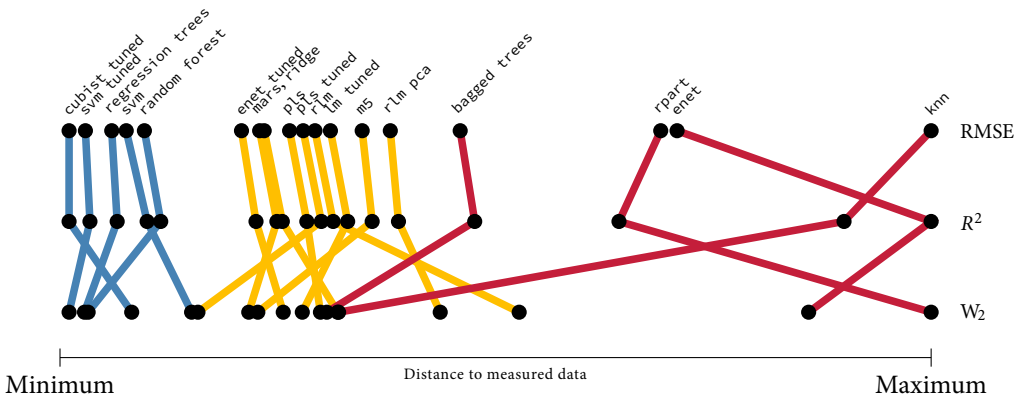


Figure 8.4: A graphical depiction of the global quality values of all models. Each dot shows the distance to the ground truth data using one of the distance measures; lower values are better. Due to their small differences, we collapsed `mars` and `ridge` into a single node. Models are linked across the different distance measures to indicate changes in the ranking. Line colours indicate the model quality, i.e. high ●, medium ●, or low ●.

value will always be achieved by a low-performance model. The numerical table makes it hard to see the differences in ranks, though. Figure 8.4 hence depicts the ranks graphically by scaling all values to $[0, 1]$ and connecting them along the different quality measures. Due to layout constraints, two models—`mars` and `ridge`—were collapsed into a single node. Overall, we can see that the ranking between RMSE and R^2 is more or less the same, except for the low-performance models.

The effects of parameter tuning are highly-dependent on the model. For example, `enet` moves from the group of low-performance models to the group of medium-performance models. The topological distance, denoted by W_2, is more sensitive for several models. For example, its ranking is more diverse in the group of high-performance models than the ranking of the other two distances. We also observe two models that are rated differently than expected. First, `knn`, which is originally considered a model of low quality, is considered to be a medium-performance model by the Wasserstein distance W_2. Second, the `bagged trees` model is also considered to be a medium-performance model and almost equal to the `pls` model. Both RMSE and R^2 consider these two models to perform differently; `knn` is considered to be among the models with the overall worst performance, for example.

Explaining changes in the ranking

The different rankings of models by the Wasserstein distance may be explained by looking at their error distributions. To this end, we define the signed error as the difference between the measured solubility value and the predicted solubility value. A value greater than zero

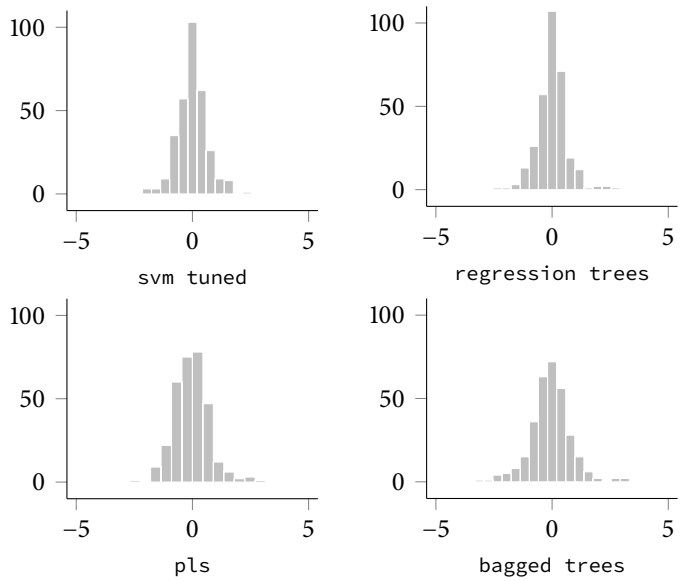


Figure 8.5: Error histograms of selected models. The histograms show the distribution of the signed error between the original solubility values and the model. A model with perfect predictive capabilities would result in a single large peak at zero. Histograms have been created using the Freedman–Diaconis rule [168].

thus indicates that the model underestimated the solubility. Figure 8.5 shows these signed error distributions for several models. We can see that two high-performance models, `svm tuned` and `regression trees`, have a very similar error distribution. They mainly differ in the amount of large errors. Likewise, the histograms for `pls` and `bagged trees` exhibit similar behaviour. The `bagged trees` model is more symmetric with respect to its errors, while the `pls` model has a tendency to overestimate the solubility of a molecule, as indicated by the slightly larger number of negative errors. The classical quality measures RMSE and R^2 are giving a disproportionate weight to these outliers, while the Wasserstein distance W_2 is less susceptible to be influenced by them.

To explain why the `knn` model is not considered to be a low-performance model, we need to go beyond established error measures. Using a scatterplot of the predicted and the measured solubility values, we can compare different models in a qualitative manner. Figure 8.6 plots the values of selected models against the ground truth values. We can see that `enet` exhibits systematic errors in its predictions, whereas `knn` merely suffers from a small amount of over- and underestimated values. As a consequence, it will be rated differently by the Wasserstein distance W_2 than the `enet` model. This demonstrates the value of focusing on *structural properties* of data—following the RMSE and R^2 would lead us to consider `knn` to be just as unsuitable as `enet`. In contrast to `enet`, the `knn` model is more similar to the original solubility

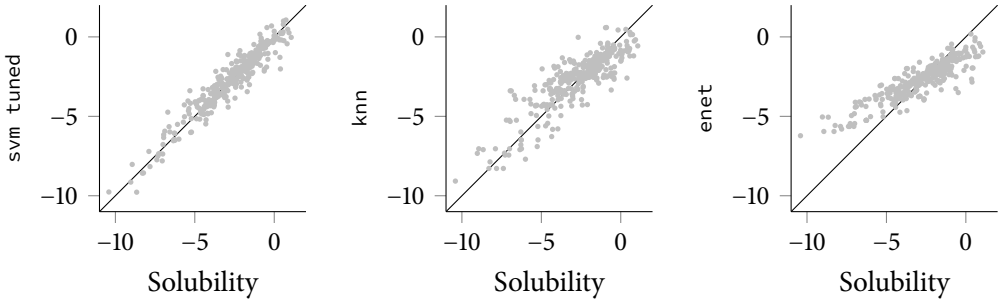


Figure 8.6: Scatterplots of selected models against measured solubility. The `svm tuned` model exhibits the best agreement with the original data. By contrast, the `knn` model shows more over- and underestimated solubility values. It performs better than the `enet` model, which exhibits a systematic error in its estimates. Nonetheless, the R^2 measure considers `enet` to perform better than `knn`.

values with respect to its geometrical–topological properties. Given a larger training data set, it is thus likely that `knn` may perform better, especially considering that its underlying approximation scheme is much simpler to calculate than most of the other algorithms.

Investigating the effects of parameter tuning

Referring back to Table 8.2, p. 207, we see that the Wasserstein distance W_2 is capable of detecting differences between all but one of the tuned models. The most significant differences occur in the performance of `enet` and `enet tuned`—which is detected by all quality measures. This model improved its ranking from one of the worst models to the top of the medium-performance group. Figure 8.7 shows histograms of the signed error to demonstrate the effects of parameter tuning. For the depicted models, parameter tuning reduces large-scale errors. Not all quality measures react in the same manner to these changes, though. For example, the improvements between `pls` and `pls tuned` are ignored by RMSE and R^2, while W_2 is capable of detecting them. The error histogram shows that this decision is justified because the `pls tuned` model exhibits a smaller amount of high-magnitude errors.

Similarity analysis using model landscapes

While useful for determining probably subgroups in the selected models, the global quality diagram is not sufficient to show inter-model distances or similarities. We thus calculate the *model landscapes* using pairwise distances for RMSE, R^2, and W_2, following the algorithm defined in Section 8.1.3. Subsequently, we briefly describe the resulting landscapes, their properties, and how to use them in order to select suitable models.

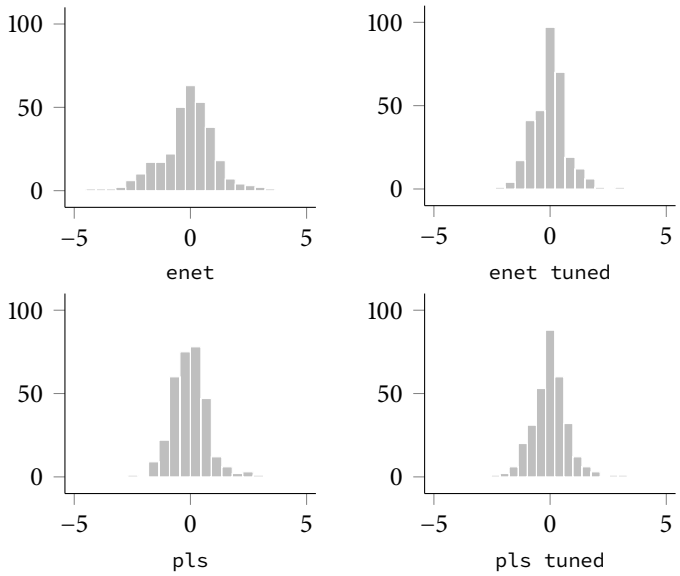


Figure 8.7: The effects of parameter tuning for several solubility models. In general, parameter tuning has a smoothing effect. The quality measures quantify this effect differently. The improvements between `pls` and `pls tuned`, for example, are only detected by the Wasserstein distance W_2. Histograms have been created using the Freedman–Diaconis rule [168].

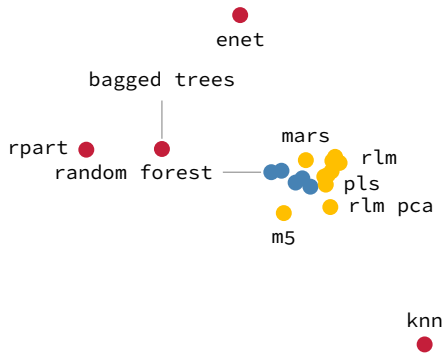


Figure 8.8: Model landscape calculated using R^2. Even though we used the transformed version of the R^2 measure, it lacks discriminative power and only yields a homogeneous 'blob' of high-performance and medium-performance models, while the low-performance models are depicted as outliers. The measure is incapable of a finer distinction between different kinds of models. For layout reasons, not all labels are shown.

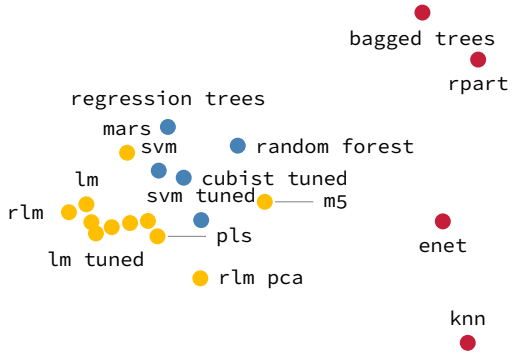


Figure 8.9: Model landscape calculated using RMSE. The measure is unable to fully discriminate between models of different performance groups. While the low-performance models such as `knn` are clearly separated from the remaining models, there is no fine-grained distinction between the medium-performance models and the high-performance models. `mars`, for example, appears to be very similar to `svm`. This is not justified by their relative performance. For layout reasons, not all labels are shown.

Model landscape for R^2 Figure 8.8 on p. 211 shows the model landscape for R^2. We can see that it does not exhibit any useful patterns for discerning different models. Even though we used the transformation as described above to make R^2 into a metric in the mathematical sense, it lacks discriminative power. Most of the models form a dense cluster with some outliers. Distances between the outliers are not meaningful, though.

Model landscape for RMSE Figure 8.9 shows the model landscape for RMSE. This landscape is already more useful. It clusters many models of medium quality. The high-performance models form a somewhat loose cluster, with several medium-quality models interspersed (`m5`, `mars`, `rlm pca`). Outliers are clearly indicated and their distance to the remaining models is indicative of their performance differences. Nonetheless, the similarities between models such as `svm` and `svm tuned` are not apparent—essentially, the distance is only capable of assessing the amount of errors between two models.

Model landscape for W_2 Figure 8.10 shows the model landscape for the Wasserstein distance W_2. In contrast to the other model landscapes, it is capable of separating the different classes of models best. Models of high and medium quality form clusters that can be distinguished, while models of low quality are positioned along the periphery of the landscape—in fact, we manually decreased their distance to the remaining models because the landscape would become too large to be shown. Similar to the RMSE landscape, the model `m5` is placed amidst the high-quality models, while `svm` is situated on the boundary of the cluster of high-quality models.

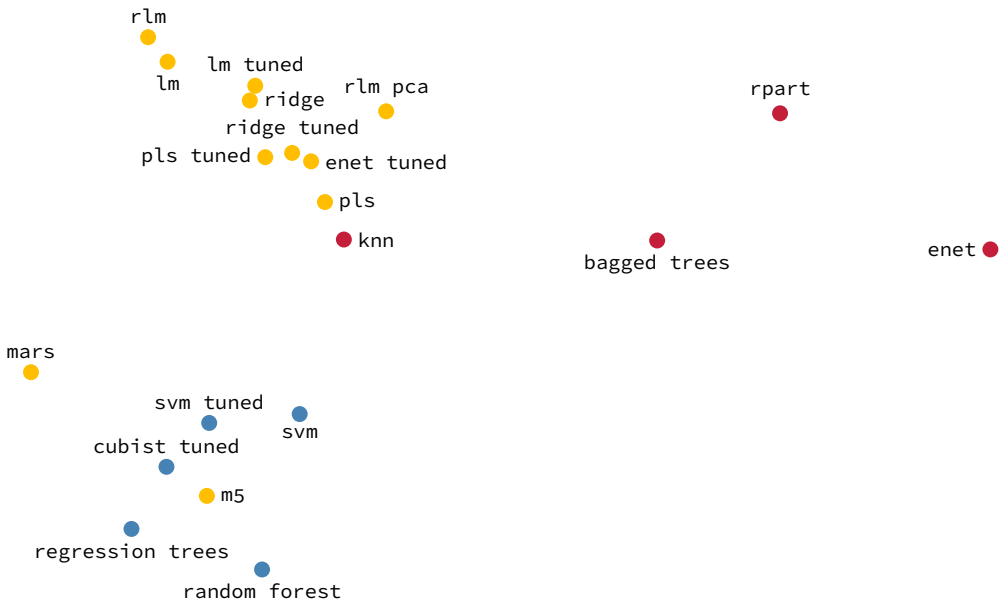


Figure 8.10: Model landscape calculated using W_2. The boundaries between structurally-different models are immediately apparent. Interestingly, `m5` is placed among other tree-based models that yield similar models. None of the other quality measures is capable of these fine-grained distinctions.

Using the model landscape The advantage of the W_2 model landscape manifests itself when selecting among competing models. If we assume that ground truth information is not available, we could first refer to the global quality plots, as depicted by Figure 8.4 on p. 208. They indicate that `cubist tuned`, `regression trees`, and `svm tuned` are good choices. However, the first two of these models take several minutes to be calculated—even on small data sets comprising a mere 400 different chemical compounds. For larger data sets, the computational efforts are thus likely to be prohibitive. The model landscape, however, indicates that `m5` is situated near these models. Since it can be calculated with less effort, it might be the better choice. Likewise, since the medium-quality models appear to be clustered densely, we may pick the ones that are easiest to calculate, such as `lm` or `rlm`—the expected small increase in predictive performance may not be worth the computational effort. The effects of applying further tuning procedures to the different models should also not be underestimated. Breiman [54] shows that averaging model predictions is a powerful tool for stabilizing them, in particular for non-linear models that are related to `m5`. This yields an explanation for the observed behaviour of `m5`. The approximation of the topology is already very good, leading to a high global quality score by the Wasserstein distance. The fact that the predictions are slightly unstable fails to be relevant when calculating W_2 due to its stability properties. Consequently, the model landscape highlights `m5` as another viable candidate.

8.2 Visualizing properties of embeddings

We have seen that even complex multivariate models can be analysed and visualized by means of their structural information, provided their predictions exist in scalar form. In the following, we want to solve the task of comparing multiple competing embeddings of multivariate data sets. This is relevant because some dimensionality reduction algorithms, such as t-SNE, are known to be capable of preserving certain aspects, for example clusters, of high-dimensional data sets. Other algorithms, such as PCA, are unable to preserve a single perceptual property well, but are capable of preserving multiple properties to some extent. Consequently, these algorithms are often only outperformed because users focus on preserving a single aspect of their data sets [257]. In this section, we will augment the landscape metaphor to depict similarities between embeddings of multivariate data sets.

At the core of the method lies the idea that to understand a data set, we should measure multiple properties instead of trying to describe the complex multivariate data directly. This is known as a *bag-of-features* [343] approach. When applying this paradigm to multivariate data, we already described several suitable functions, which we referred to as *data descriptors*. In Chapter 7, Section 7.5, p. 170 ff., we encountered a workflow that permits us to quantify how well a set of embeddings preserves a *single* property of the data, e.g. its density. Subsequently, we will see how to extend the persistent homology algorithm in order to obtain information about the topological distances between the original data and *multiple* data descriptors. Next, we will explain how to create a landscape using this paradigm.

8.2.1 Depicting multiple data descriptors

In the following, we require the existence of an unstructured point cloud—the *reference point cloud* $\mathcal{P}$—and k *derived point clouds* $\mathcal{P}'_1, \mathcal{P}'_2, \ldots, \mathcal{P}'_k$. A derived point cloud is typically calculated using a dimensionality reduction method. However, it may also be another time-step of a time-varying point cloud, for instance. We propose the following workflow for creating a *data descriptor landscape*:

1. Calculate a Vietoris–Rips complex $\mathcal{V}_\epsilon$ for the reference point cloud $\mathcal{P}$. The simplicial complex $\mathcal{V}_\epsilon$ will remain fixed for the subsequent analysis.

2. Calculate each data descriptor $f_1, f_2, \ldots, f_k$ on each derived point cloud $\mathcal{P}'_j$. This yields a set of scalar functions $\{f_{1j}, \ldots, f_{kj}\}$.

3. Use every scalar function as weights for $\mathcal{V}_e$ and calculate its persistent homology. This results in a set of persistence diagrams $\{\mathcal{D}_{1j}, \ldots, \mathcal{D}_{1k}\}$. For the sake of a simpler notation, we do not index the individual dimensions of the persistence diagrams further. Each persistence diagram describes the geometrical–topological features of a data descriptor on a derived point cloud.

4. We perform the same procedure for the original data set, yielding a set of persistence diagrams $\{\mathcal{D}_1, \ldots, \mathcal{D}_k\}$. These diagrams describe the geometrical–topological features of a data descriptor on the reference point cloud $\mathcal{P}$.

5. Calculate the Wasserstein distance W_2 between the reference persistence diagrams and the derived persistence diagrams. This results in an $n \times k$ matrix M of these distances. We have

$$m_{ij} = \mathrm{W}_2(\mathcal{D}_{ij}, \mathcal{D}_j), \tag{8.12}$$

   i.e. the i^{th} row consists of all topological distances of the i^{th} derived point cloud $\mathcal{P}'_i$ to the reference point cloud $\mathcal{P}$, measured using some data descriptor f_j.

6. Perform a PCA of M in order to obtain coordinates in $\mathbb{R}^2$. Use *star glyphs* [384], as described in Chapter 2, Section 2.3, p. 16, ff., on every resulting position in order to visualize the data descriptor.

The matrix M that is used in this workflow affords an intuitive interpretation. As its entries are distances, smaller values indicate a better fit. More precisely, this means that the topology of the data descriptor is very similar on both the original point cloud and the derived point cloud. If the derived point cloud is an embedding of the original point cloud, for example, topological similarity implies that the property measured by the data descriptor—such as density—has been preserved in the embedding. To further emphasize this point, we colour-code the glyphs in the *data descriptor landscape* by their Euclidean norm using a continuous colour map . Darker colours correspond to larger norms, which in turn indicate that the row vector contains large values for at least one data descriptor. This means that the original point cloud and the derived point cloud differ in at least one aspect by a large extent. Similarity is thus indicated in two different ways by the landscape. First, spatial proximity indicates topological similarity with respect to multiple data descriptors. Second, similar glyphs reveal a similar distribution of errors in the data descriptors.

For the subsequent analysis, we use the three data descriptors from Chapter 7, Section 7.5, p. 170 ff., which measure *density*, *eccentricity*, and *linearity*. These three properties are known to be relevant when analysing an embedding [242, 357]. The workflow is sufficiently generic to permit other descriptors, though.

8.2.2 Results

In the following, we first present two examples in which we use the data descriptor landscape to analyse dimensionality reduction methods. For the synthetic faces data, we will see how different algorithms yield similar embeddings, whose similarity is indicated by the landscape. Furthermore, we will observe how the stability of some algorithms varies with respect to small perturbations of the neighbourhood size parameter. For the climate data, we will detect that many non-linear methods are incapable of generating a suitable embedding. Moreover, we will study how changing the neighbourhood size parameter affects the shape of an embedding.

Synthetic faces

We have already encountered the synthetic faces data in Section 7.9.1, p. 184 ff., where we focused in particular on density preservation in embeddings. Even though Tenenbaum et al. [359] showed that Isomap is able to capture phenomena that cannot be adequately captured by linear dimensionality reduction methods, we saw earlier that Isomap may become unstable when changing the number of neighbours. In the following, we will see that this instability also manifests itself in different data descriptors, which is clearly indicated in the data descriptor landscape. Furthermore, we will see how spatial similarity in the data descriptor landscape highlights different dimensionality reduction algorithms that yield very similar embeddings, both in terms of appearance and in terms of preserved properties. Currently, there are no other state-of-the-art techniques that are capable of providing this—or a related—overview.

Figure 8.11 depicts the data descriptor landscape, with several selected labelled embeddings. Since all methods—except PCA—have a parameter k for tuning the neighbourhood, we indicate the value of the parameter within the label of the embedding. We first note that Isomap nodes are placed at very different positions with very different glyphs. This implies that the embeddings differ in the sense that they do not preserve the data descriptors to similar extents. Already the embeddings for $k = 8$ and $k = 9$ start to become unstable. As we have already seen in Chapter 7, Section 7.9.1, p. 184 ff., higher values for k yield increasingly bent embeddings which cannot preserve the data descriptors. This is indicated in the data descriptor landscape by both glyph colour and glyph shape. Figure 8.12 shows several Isomap embeddings for varying values of k.

We observe a further feature of the landscape when we take a look at the group of nodes in the middle. It is dominated by instances of HLLE and LTSA, two algorithms that use completely different mathematical models but arrive at very similar embeddings. Varying their k parameters has less dramatic effects than for the Isomap algorithm. We observe that most embeddings stay in the same region of the landscape, which corresponds to almost unchanged data descriptors. Only when choosing a very large value for k do we observe

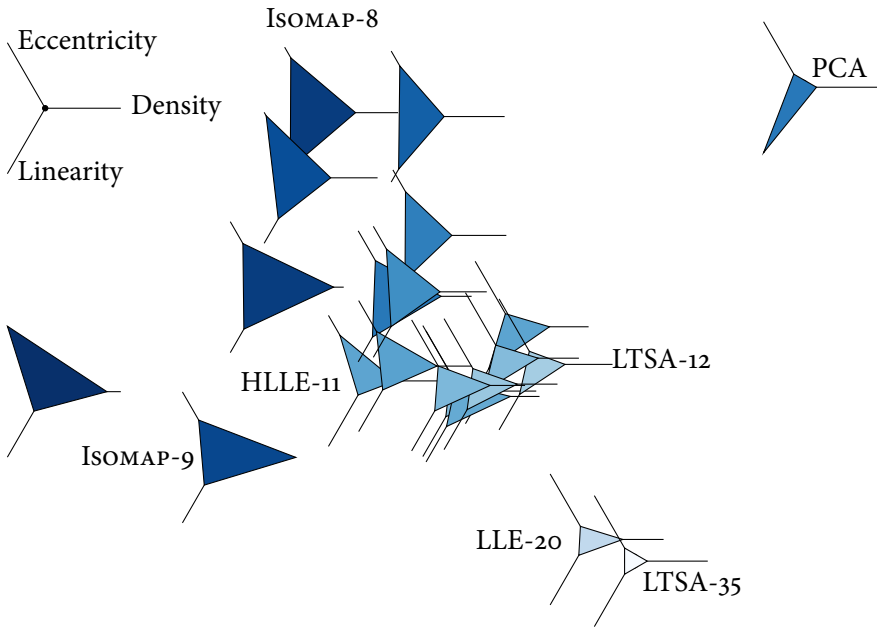


Figure 8.11: Data descriptor landscape of the synthetic faces data. Every glyph represents a different embedding. The legend in the upper-left corner indicates the order of the different data descriptors in the glyph, while the colours encode the norm of the feature vector, as described in Section 8.2.1. For layout reasons, not all labels are displayed. It is interesting to note that Isomap embeddings suffer from instabilities. Increasing k by one may already result in a markedly different embedding. Moreover, the central region of the landscape is dominated by HLLE and LTSA embeddings, whose embeddings are very similar even though the algorithms are completely different.



Figure 8.12: Example embeddings of the synthetic faces data. Embeddings start to bend already when increasing k by one. We experienced similar behaviour in Chapter 7, Section 7.9.1, p. 184 ff., where we only focused on preserving density. The glyphs in the data descriptor landscape indicate that embeddings with larger neighbourhood parameters k are capable of preserving eccentricity and linearity slightly better.

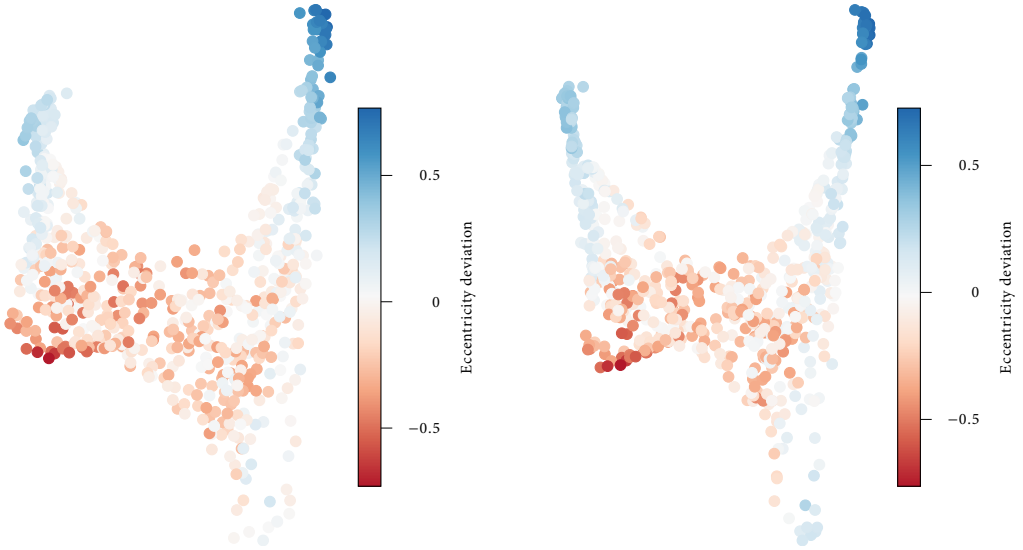


Figure 8.13: Eccentricity deviations for selected embeddings of the synthetic faces data. Blue values indicate that the eccentricity of the embedding is *higher* than the eccentricity of the original data. Red values indicate that the eccentricity of the embedding is *larger* than the eccentricity of the original data. Regions that appear to be stretched or squeezed in the embedding tend to exhibit a larger amount of eccentricity deviations. This increases the error in the eccentricity data descriptor.

differences in the embedding. The glyphs of both types of embeddings indicate that neither the density nor the eccentricity descriptor are retained to a sufficient extent. Density and local distances are thus not very trustworthy in these embeddings. From the visualizations of two of the embeddings, as shown in Figure 8.13, we see that the deviations are caused by the stretching that occurs in both embeddings. By calculating the ratio between the eccentricity in the embedding and the eccentricity of the original data, we observe the largest deviations in these areas.

The most outlying glyphs are also those with the smallest norms, indicating that all data descriptors are being retained similarly well. Figure 8.14 shows several suitable embeddings of the synthetic faces. Although LTSA exhibits the least amount of errors in all data descriptors, PCA may be a better choice for these data, even though it is incapable of representing the linear structures well. Note that the PCA embedding resembles the MDS embedding of the synthetic faces depicted in Figure 7.21, p. 185, because we used Euclidean distances for both calculations [49, pp. 524–526]. This sort of analysis process goes beyond the capabilities of state-of-the-art methods. Our method permits users to immediately see how different algorithms preserve multiple salient aspects of their data—or fail to do so.



Figure 8.14: Suitable embeddings of the synthetic faces data. These three embeddings exhibit the least amount of errors in all data descriptors. In particular, LTSA is capable of retaining all descriptors equally well—at the expense of a larger runtime of approximately 10 s in comparison to the 0.5 s for the PCA embedding.

Climate data

Here, we return to the multivariate data set from climate research that we previously introduced and analysed in Chapter 7, Section 7.9.3, p. 189 ff., with respect to its density. We will subsequently focus on the trustworthiness of certain attributes, such as *linear structures*, in different embeddings. Just as for the previous analysis of this data set, we ignore the geographic location at which the different measurements were taken because we are interested in an embedding of the overall attribute space.

Figure 8.15 on p. 220 shows the data descriptor landscape. We observe a clear separation between linear and non-linear dimensionality reduction methods, both in terms of glyph placement, glyph shape, and glyph colour. Except for *stochastic proximity embedding* (SPE), the tested non-linear methods are incapable of preserving the data descriptors to a sufficient extent on this data set. Recalling the results from Chapter 7, Section 7.9.3, p. 189 ff., this shows the benefits of analysing multiple descriptors at once. If we were to focus on density solely, for example, parameter tuning for *locally linear embedding* (LLE) would result in suitable embeddings. By including more information about other properties present in the data, we see that the good density conservation is at the expense of other structural properties that are not preserved.

Focusing on the suitable embeddings, we see that *factor analysis* (FA) misrepresents all data descriptors to some extent. By increasing the number of iterations n of this algorithm, the quality of the embedding starts to increase. In the data descriptor landscape, we added the amount of iterations to the label of this algorithm. PCA, on the other hand, misrepresents density and linearity, while SPE exhibits errors in eccentricity and linearity. In contrast to the other non-linear methods, these errors are comparatively small though. Figure 8.16 shows the corresponding embeddings. Through the glyphs, we can see that the linear structures

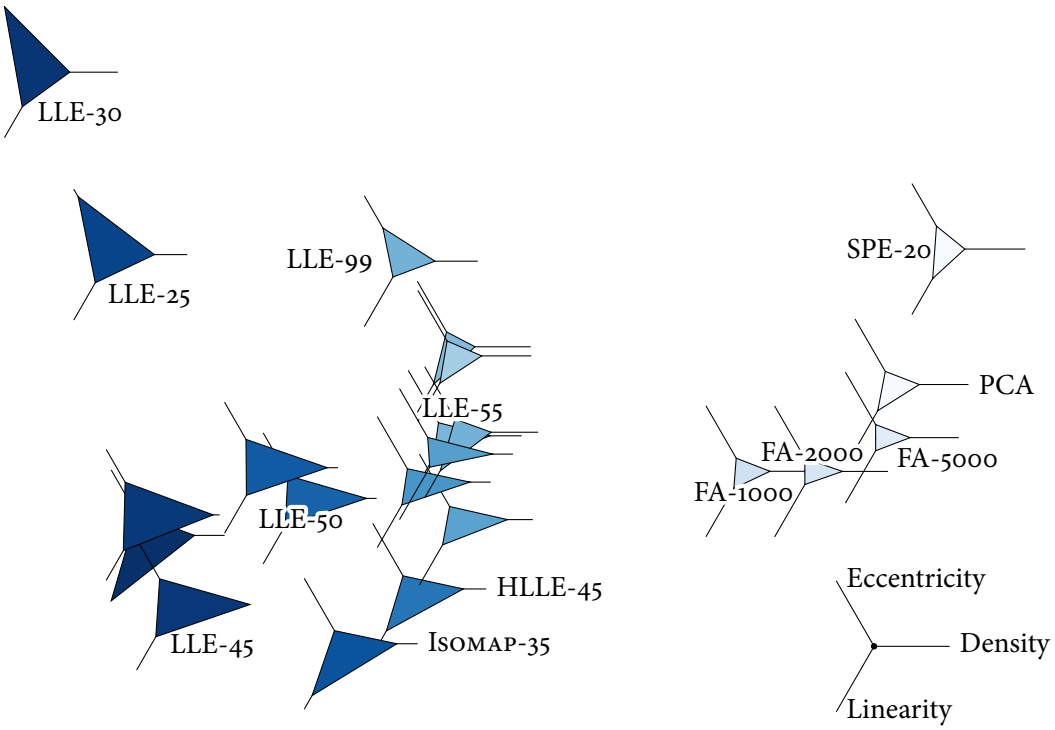


Figure 8.15: Data descriptor landscape of the climate data. For layout reasons, not all labels are displayed. The most prominent feature in this landscape is the split between linear and non-linear methods. Apart from SPE, only linear dimensionality reduction methods are capable of preserving multiple data descriptors.

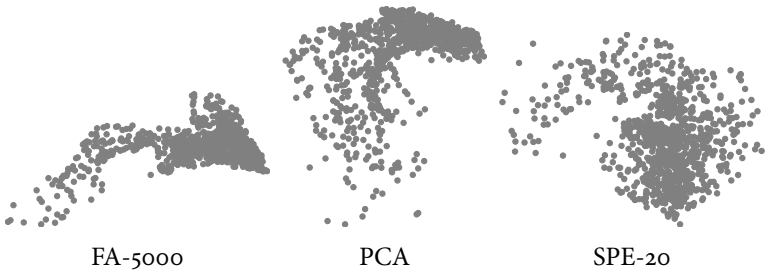


Figure 8.16: Suitable embeddings of the climate data. All embeddings exhibit a similar shape and are incapable of preserving density to some extent.

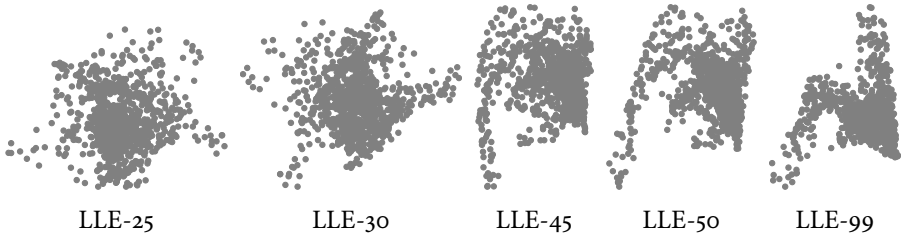


Figure 8.17: Instability of LLE embeddings for the climate data. Upon increasing the number of neighbours k for the LLE algorithm, the embedding varies its shape and drastically changes the appearance of the data set.

that appear so prominently in the FA embedding are a salient feature of the data and not a structural illusion. Those structures are also hinted at in the other embeddings, but not in the clearly-defined manner as in the FA embedding.

Again, we can employ the data descriptor landscape to help evaluate the effects of parameter tuning. The embeddings generated by LLE, for example, drastically change their form and shape when increasing the number of neighbours k. Figure 8.17 shows numerous embeddings. With higher values for k, the separation into a central structure with a 'flare' becomes more pronounced. All these embeddings do not preserve eccentricity in the data very well. We thus conclude that either SPE or FA are suitable choices for embedding these data, followed by PCA. The latter algorithm does not preserve density and linearity similarly as well as the other methods, though.

8.3 Discussion

In this chapter, we encountered the powerful *landscape metaphor* to analyse multivariate data under either a single aspect or under multiple aspects. We first presented a topology-driven embedding for improving the comparative analysis of competing models in the context of regression analysis. To this end, we defined a global persistence-based quality measure that complements existing measures with its proven stability properties. Following this, we presented *model landscapes*, a visual representation of pairwise model dissimilarities. We used different model landscapes for exploring relationships between models and observed that the novel quality measure, based on the second Wasserstein distance W_2 between persistence diagrams, is significantly more sensitive for discriminating between different models than the state-of-the-art measures RMSE and R^2.

Furthermore, we extended the model landscape to analyse multivariate data under several aspects at the same time. Based on the data descriptor concept and workflow we introduced in Chapter 7, we developed the *data descriptor landscape*. This glyph-based visualization

permits the rapid comparison of multiple multivariate data sets under different quality aspects. We used data descriptor landscapes to quickly detect structurally and perceptually similar embeddings. Since the data descriptor landscape is an information-rich visualization in 2D, this even works for high-dimensional embeddings that cannot be directly visualized. Moreover, we judged the suitability of embeddings with respect to the preservation of relevant properties of a data set. The data descriptor landscape permitted us to assess the stability of parameter choices, thereby helping increase the trustworthiness of dimensionality reduction methods. We demonstrated the capabilities of this novel visualization on several data sets.

Model landscapes: Extensions & future work

There are several potentially useful enhancements for model landscapes. For improving the selection and comparison of models, the model landscape should include a measure of uncertainty—if different models yield extremely varying estimates for a given region of the input data, this should be reflected in the model landscape. For a certain class of models, Gosink et al. [180] were able to obtain a measure of their uncertainty through averaging. With more generic model algorithms, obtaining this information is likely to be more complicated because the correct domain of the input function is often unknown.

Somewhat orthogonally, the model landscape should also incorporate uncertainty about the persistent homology calculation itself. Here, we could make use of recent advances in the statistical analysis of persistence diagrams [90, 163], for example. Another aspect of future research thus involves attempting to understand the domain of the input data. The domain approximation approach by Gerber et al. [176], based on Morse–Smale complexes, could be a useful starting point.

The model landscape could also be used to detect unsuitable splits into training and test data. By performing the model landscape calculation for multiple splits of a data set, we could derive and visualize *confidence regions* for each model in the landscape. Small confidence regions correspond to models whose behaviour remains stable and consistent regardless of different splits, whereas large confidence regions could indicate that a model requires more training samples to improve its predictive capabilities.

Data descriptor landscape: Extensions & future work

Since the data descriptor landscape is intended to evaluate dimensionality reduction methods, it could easily be integrated in established user-centric evaluation systems, such as the DimStiller framework [209]. We also think that the observed behaviour of some dimensionality reduction algorithms necessitates an investigation of different synthetic and real-world data

sets on a larger scale. Analysts are in need of knowing the theoretical and practical limits of their method, as well as the models they are based on to make an informed choice.

The data descriptor landscape is sufficiently generic to be applied in different contexts. A particularly interesting scenario would involve the characterization of numerous multivariate data sets that vary over time. For such data, the data descriptor landscape will result in groups of glyphs that represent data sets with similar shapes—the underlying assumption being that a similar shape corresponds to similar behaviour represented in a data set. Especially in a time-varying context, additional information in the form of *trajectories* could be added to the landscape. Previous work by Bach et al. [19] shows that this has certain perceptual advantages. It could potentially enable analysts to quickly sift through larger amounts of multivariate data sets and perform a coarse pre-classification by means of the resulting trajectory profiles.

Another aspect for future work involves the integration of *multidimensional persistence*. Instead of using multiple scalar-valued filtrations, this concept would permit us to describe the behaviour of vector-valued functions on our data. This approach promises to yield more detailed information about correlations between different data descriptors. Multidimensional persistence was introduced by Carlsson and Zomorodian [71] to extend persistent homology to vector-valued functions. A subsequent publication [70] contains an initial attempt at a computation algorithm. Unfortunately, it turns out that a full computation or description of multidimensional persistent homology is not possible for now; only a very weak part of the data, the *rank invariant*, may be calculated. However, the author is convinced that the close relation of multidimensional persistence to *multidimensional size theory* [84] could be exploited in order to advance research in this direction. Cerri et al. [85] recently showed that multidimensional persistence spaces may be compared in a stable manner—similar to the different metrics we already encountered for persistence diagrams. Still, low-dimensional approximations of multidimensional persistence are being used to analyse real-world data sets [380, 395] and the results seem to be promising thus far.

9 Assessing & visualizing clusterings

In this chapter, we focus on clustering algorithms again. We have seen in previous chapters that clustering plays an undeniably important role in any workflow that involves *exploratory data analysis* (EDA). Here, we now pose the question of how *trustworthy* the results of a clustering algorithm are. Since clustering helps users gain a mental model of even the most complex multivariate data sets, there is a need for an external evaluation of the results. Nowadays, production-quality clustering libraries such as SCIKIT-LEARN [289] make *obtaining* a clustering easy—the challenge lies in *assessing* whether the clustering is describing relevant information. This chapter is based on a previous publication by the author [314].

✦

In this chapter, we will take a look at how to assess and visualize the results of complex clustering algorithms. This endeavour naturally splits into two—not necessarily independent—tasks. First, we need to figure out how to choose a suitable clustering algorithm. This choice only depends on the data. A well-known maxim in data mining states:

> There is no best clustering algorithm. [...] When there is a good match between the model and the data, good partitions are obtained. [211]

Consequently, choosing a suitable algorithm involves a comparison between structures in the data and structures in the clustering. This leads to the second task, namely the number of clusters to look for. Most clustering algorithms, such as the well-known k-means algorithm [211], use a single parameter k that determines the number of clusters into which a data set is to be partitioned. Other algorithms, such as DBSCAN [160], may feature even more parameters, but for now, we only focus on k. How to find suitable values for k is still actively debated within the clustering community. A common approach is to run different clustering algorithms with varying parameters. During each run, a *clustering validity index* is evaluated, and k is selected such that the index shows the 'best' value. The literature pertaining to clustering validity indices is vast and ranges from simple indices such as the *Dunn index* [139] to more complex pointwise measures such as the *silhouette coefficient* [320]. However, while the indices are valuable for comparing different clusterings from the same algorithm [187], their use for EDA is somewhat limited. It turns out that complex cluster geometries often lead

to unstable results so that a suitable k fails to be found even for comparatively simple data sets such as the 'Iris flower' data [402]. Furthermore, existing clustering validity indices are unable to assess individual clusters of a clustering without referring to labels, which are often unavailable in real-world data sets. We will encounter examples of these issues in this chapter.

To support users in choosing a suitable clustering algorithm and a suitable number of clusters, this chapter presents two novel visualization techniques for clusterings. The *clustering similarity graph* provides a global view of multiple clustering by arranging them using a similarity measure. Individual clusterings are colour-coded according to their suitability of preserving prominent structures in the data. Moreover, the *cluster map* yields a local view of the individual clusters that make up a clustering. Every cluster is depicted using a highly-informative glyph. All glyphs are arranged along a common reference embedding of the data to simplify orientation. This makes it possible for users to see how a given clustering partitions their data. In turn, this information permits the comparison of clusters among each other and among different clusterings of the data. Both visualizations employ a novel clustering assessment measure based on persistent homology. Our measure provides a well-defined way of comparing different clusterings, both on a global and on a local level. We show that our measure is robust and stable in the presence of noise. Furthermore, by analysing numerous data sets using state-of-the-art clustering validity measures, we demonstrate that our topology-based measure is capable of assessing even complex cluster shapes. Our measure turns out to be on par with the best existing techniques and in many cases, it even outperforms them.

9.1 Related work

The methods described in this chapter are situated on the intersection between visualization, topological data analysis, and data mining. We will therefore first take a look at previous work and how it relates to our approach. In general, the visualization of *different* clustering algorithms has been largely ignored in the visualization community so far. Recent work by Zhang et al. [404] started to remedy this by showing how clusters change under geographical variations of multivariate data.

Evaluating clustering algorithms

The amount of available clustering algorithms is staggering [397]. Clustering remains one of the most important techniques for making sense of multivariate data sets, especially given the existence of many robust and high-quality clustering libraries for different programming languages [221, 289]. However, the question of what constitutes a good cluster still remains unanswered, as it is highly-dependent on the particular data [211]. The complexity of real-world

data necessitates the use of different internal measures for assessing a given clustering [139, 320, 347]. An in-depth comparison [187] shows that they are unstable in the presence of noise, overlapping cluster boundaries, or complicated geometries [10]. We shall encounter several examples in Section 9.2.5, where we calculate existing validity indices on several test data sets. The results indicate that even simple cluster geometries may pose problems for the assessment. By contrast, the quality measures that we introduce in this chapter do not suffer from the same set of limitations.

Clustering & visualization

So far, the visualization community has used clustering merely as an auxiliary tool for making sense of multivariate data. For example, some methods evaluate clusters within auxiliary visualizations of data, such as parallel coordinate plots [201] or scatterplots [155]. However, there is also a great interest in helping users understand the results of a given clustering algorithm. The most prominent tool for this purpose is the *hierarchical clustering explorer* [333], which enables biologists to interact with different hierarchical clusterings of microarray experiment data. Various model-based indices and auxiliary visualizations aid in understanding not only the data space but also the decisions made by the clustering algorithm. Similarly, a framework by Lex et al. [244] lets users modify the results of clustering algorithms by e.g. splitting data into subgroups, which are then clustered separately. While this helps detect relations that would otherwise remain obscured by classical methods, it still requires a small 'leap of faith' concerning the trustworthiness of the clustering algorithm.

We find several similar ideas in visual analytics tools. Nam et al. [280], for example, let users 'sculpt' clusters by changing the relevance of different attributes in the data, while using several auxiliary visualizations to make sense of the current clustering. Schreck et al. [327] embed clustering analysis in a general visual analytics workflow. Their system permits modifying clustering results as well as verifying their validity, but forgoes traditional clustering algorithms in favour of *self-organizing maps* [223]. Hence, it does not permit the comparison of multiple clustering results. Tatu et al. [358] support the clustering process by pre-selecting interesting subspaces—with respect to fluctuations in density, for example—in the data, which are then clustered using hierarchical clustering. Users may interact with dissimilar subspaces and learn their structures via interaction. This approach is somewhat orthogonal to our approach in that it helps explore patterns in the data prior to applying any clustering algorithms. Pilhöfer et al. [294] developed a method for re-ordering categorical variables in order to improve visualizations of multiple clusterings. This permits tracking similarities of partitions over different clusterings. In contrast to this, our method focuses more on exploring the shapes of individual clusters in a single clustering.

9.2 Methods

The main drawback of existing quality measures for clusterings is their lack of a useful baseline. We propose a new way of looking at clusterings, based on the topological information inherent to the data. This permits us to assess a clustering both on the global and on the local level. The foundation for this assessment is again the notion of a *data descriptor*, which we already encountered in Chapter 7, Section 7.5, p. 170 ff., in the context of analysing embeddings of high-dimensional data sets. Briefly put, our algorithm consists of the following steps:

1. Choose a data descriptor function and use it to calculate persistent homology on the original data. This yields a persistence diagram $\mathcal{D}_{\text{Original}}$.

2. Given a clustering of the data, extend the partition onto the Vietoris–Rips complex that was used to calculate persistent homology.

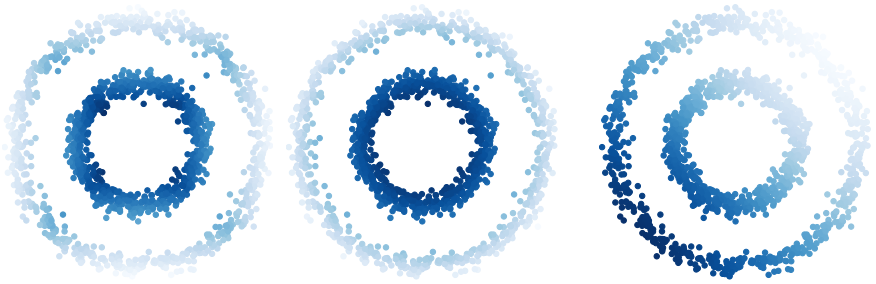
3. Use this information to generate a set of persistence diagrams. Each persistence diagram describes the geometrical–topological features of the data descriptor function on a certain cluster of the given clustering.

4. Compare the persistence diagrams against the original persistence diagram. In particular, it needs to be checked whether all geometrical–topological features of the original persistence diagram are still present in the persistence diagrams of individual clusters.

At the outset, this bears many similarities to the previously-encountered methods. Here, the challenge is to compare persistence diagrams without using existing topological distance measures. We shall see later why these measures are not applicable here. Furthermore, in contrast to previous chapters, we will use a modified variant of persistent homology—*extended persistence*—that simplifies the assessment of similarities between different diagrams.

9.2.1 Choosing a data descriptor

We have seen numerous examples of useful data descriptors in Chapter 7, Section 7.5, p. 170. In the clustering context we found the *eccentricity* shape descriptors to be particularly useful. If not mentioned otherwise, we will be using this descriptor with $p = 2$ through the remainder of the chapter. Previous publications [67, 253] already showed that this descriptor is capable of describing salient structures in high-dimensional data sets. In general, the utility of a particular data descriptor function largely hinges on its discriminative properties. Biasotti et al. [42] investigate to what extent these properties are present in certain functions. The eigenfunctions of the graph Laplacian matrix, for example, are generally suitable for multivariate data analysis [32, 342]. If the input data set permits it, kernel regression estimators such as the



Eccentricity ($p = 2$) First SVD coordinate Second SVD coordinate

Figure 9.1: An illustration of different data descriptors for clustering analysis. The colour indicates the data descriptor value for the corresponding point. Since the data set is not high-dimensional, the eccentricity data descriptor and the SVD data descriptor for the first coordinate are slightly similar. Each of these descriptors may be used in the subsequent analysis.

Nadaraya–Watson estimator [279] may also prove to be useful. Standard matrix decomposition algorithms, such as the *singular value decomposition* (SVD), also yield useful information. For example, we can decompose the distance matrix associated with a clustering—as defined in Section 9.2.5—and obtain scalar values by using the coordinates of the first or second singular vector. Figure 9.1 illustrates several data descriptors for clustering analysis.

9.2.2 Extended persistent homology

Recalling the calculations of persistent homology from the previous chapters, we know that the pairing process underlying the calculation of persistent homology is asymmetrical. For example, when we calculate 0-dimensional persistence diagrams, at least one simplex—the one corresponding to the minimum function value—will remain unpaired. By contrast, all other simplices are paired. As a consequence, we have to either think about which persistence value to assign to these unpaired points or remove them altogether from the analysis. A partial solution for this issue is given by *reduced homology* [141, p. 83], but it applies to 0-dimensional persistence diagrams only. Provided we have function values for the 0-dimensional simplices available, we can use *extended persistence*, a more complicated technique for ensuring the symmetry of a pairing.

Extended persistent homology of a 2-manifold

Figure 9.2 illustrates the calculation of extended persistent homology for a 2-manifold. We assume that we are given a 'natural height function' f that measures distance from the plane along which the manifold is oriented. In this setting, there is a large overlap between persistent homology and Morse theory [270, 273]. We already encountered Morse theory briefly in Chapter 2, Section 2.6, p. 18 ff., so we know that the interesting features of this height function

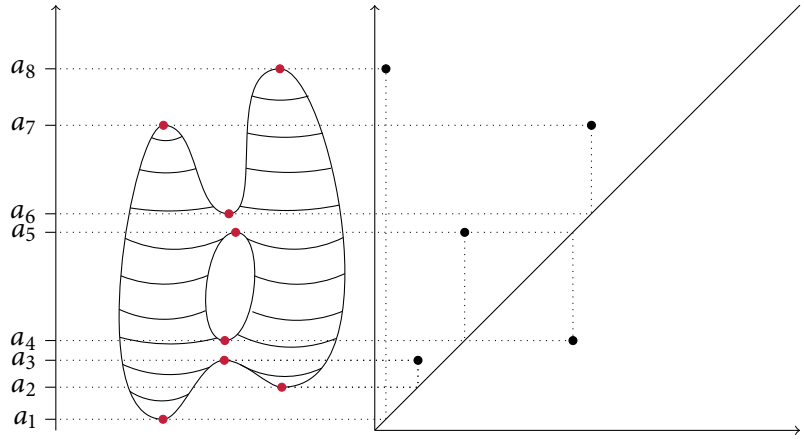


Figure 9.2: An illustration of the extended persistent homology calculation. When calculating ordinary persistent homology, at least the critical point a_1 would remain unpaired. Extended persistent homology, on the other hand, results in all critical points being paired.

are given by the *critical points* of f. These are all the points where the gradient of f vanishes, i.e. $\nabla f = 0$. In 2D, a full classification of the different types of critical points exists—we only have local minima, local maxima, or saddles. In the example above, a_1, a_2, and a_6 are local minima. Similarly, a_3, a_7, and a_8 are local maxima. Finally, a_4 and a_5 are saddle points; here, f can both increase or decrease within an arbitrarily small neighbourhood.

As usual in Morse theory, we analyse the connectivity of the sublevel sets of f. The subsequent algorithm works analogously for the superlevel sets. Traversing the sublevel sets of f and calculating persistent homology requires pairing the critical points. A minimum always creates a new connected component in the corresponding sublevel set. A saddle either merges two connected components—thereby destroying the one with the larger height—or creates a new hole (but not both). Last, a local maximum destroys a hole by closing it, while the global maximum creates a new void. Following the ordinary persistent homology calculation as described by Algorithm 5 on p. 62, we observe that we cannot pair the global minimum at a_1, the two saddles at a_4 and a_5, and the global maximum at a_8. The global minimum, for example, creates a new connected component that is never destroyed by a merge. To resolve the asymmetry of this situation, we need to pair the remaining points in a consistent manner. We thus pair a_1 with a_8 in order to denote the range of f on the manifold. Next, for symmetry reasons we pair the two saddles with each other, so that we get both (a_4, a_5) and (a_5, a_4) as points in the persistence diagram. This notion of a pairing of features was introduced by

Cohen-Steiner et al. [103] in the context of measuring the elevation above a surface. It yields the *extended persistence diagram* of f. The advantage of using extended persistent homology is that we obtain finite persistence values for all topological features. This makes assessing the individual parts of a clustering easier. We only require a scalar-valued function on the data, but we have seen plenty of examples of these functions in the previous chapters.

Calculating extended persistent homology

The calculation of extended persistent homology is slightly more involved than the algorithms we have previously encountered. In the following, we assume that we have a simplicial complex K and a scalar function $f\colon \operatorname{vert} \mathrm{K} \to \mathbb{R}$ on its vertices. We require the function values of f to be distinct. This may always be achieved in practice by *simulation of simplicity* [151], numeric perturbation, or a consistent ordering of values. The scalar function f induces a partition of K into its *lower stars* and its *upper stars*.

Definition 9.1 (Star of a simplicial complex). Given a simplicial complex K and a simplex $\tau \in \mathrm{K}$, the *star* of τ contains all cofaces of τ, i.e.

$$\operatorname{St} \tau := \{\sigma \in \mathrm{K} \mid \sigma \subseteq \tau\}, \tag{9.1}$$

which is in general not a simplicial subcomplex. For a vertex $v \in \operatorname{vert} \mathrm{K}$, we may expand the previous definition and define the *lower star* of v as the set that contains all simplices in K for which v is the vertex with the highest function value. The *upper star* is defined analogously. Formally, we have:

$$\operatorname{St}^- v := \{\sigma \in \operatorname{St} v \mid \{x\} \in \sigma \Rightarrow f(x) \leq f(v)\} \tag{9.2}$$

$$\operatorname{St}^+ v := \{\sigma \in \operatorname{St} v \mid \{x\} \in \sigma \Rightarrow f(x) \geq f(v)\} \tag{9.3}$$

Both the lower star St^- and the upper star St^+ result in a well-defined partition of K because every simplex has a unique lowest or highest vertex by assumption.

To use the partitions defined by the stars, we extend the simplicial complex K with a 'dummy vertex' v. For each simplex $\sigma \in \mathrm{K}$, we then add $\sigma \cup \{v\}$ to the simplicial complex. This operation is known as calculating the *cone* of a simplicial complex. The cone of K is a valid simplicial complex on its own. We use the lower stars to obtain a filtration of the original simplicial complex. To this end, we sort K according to the lower stars of its vertices $v_0, v_1, \ldots$, with $f(v_0) < f(v_1) < \ldots$ for the reasons explained above. Next, we extend this filtration with all the simplices of the cone. This requires sorting the cone according to the upper stars of the vertices $v_n, v_{n-1}, \ldots$, with $f(v_n) > f(v_{n-1}) > \ldots$ to ensure symmetry. Finally, we calculate the

persistent homology of this *extended filtration*. Adding the cone ensures that *every* homology class is eventually paired. To obtain the *extended persistence diagram* as described above, we ignore all persistence pairs in which the dummy vertex appears. Algorithm 17 gives a short overview of the calculation. Extended persistent homology has interesting symmetry and duality properties [103], which we only briefly touch upon. For example, the first part of the extended filtration—which involves the lower stars—yields the same persistence pairs that we obtain using the standard persistent homology algorithm [141, pp. 163–165].

Algorithm 17: Extended persistent homology calculation

Require: Simplicial complex K, scalar function $f\colon \operatorname{vert} \mathrm{K} \to \mathbb{R}$

1: Sort vertices $v_0, v_1, \ldots$ of K such that $f(v_0) < f(v_1) < \ldots$ holds.
2: Sort K according to the lower stars of its vertices.
3: Calculate the *cone* K' of K.
4: Sort vertices $v_n, v_{n-1}, \ldots$ of K' such that $f(v_n) > f(v_{n-1}) > \ldots$ holds.
5: Sort K' according to the upper stars of its vertices.
6: Calculate persistent homology of K and K'.

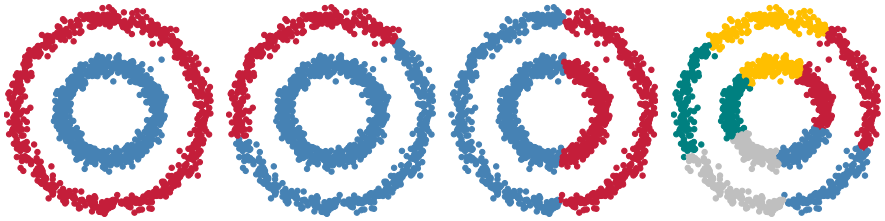
9.2.3 Total persistence

So far, we have only seen how to obtain an extended persistence diagram from our input data. In Chapter 4, Section 4.6.1, p. 68 ff., we already discussed algorithms for quantifying the similarity between persistence diagrams under the assumption that they describe related phenomena. However, for assessing a clustering, we need to quantify the similarity between different *parts* of a persistence diagram—in essence, we are solving a jigsaw puzzle. As a consequence, we need summarizing measures that quantify the amount of geometrical–topological variation encoded by a persistence diagram. A very useful summary statistic in this context is given by the *total persistence* of a persistence diagram. We already encountered a general description of this measure in Definition 4.28, p. 74. Here, we want to focus on one particular variant.

Definition 9.2 (Total persistence). Given the persistence diagram $\mathcal{D}_f$ of a function f, its total persistence is defined as the sum of all squared persistence values, i.e.

$$\operatorname{Pers}(\mathcal{D}_f) := \sum_{(c,d)\in\mathcal{D}_f} \operatorname{pers}(c,d)^2, \tag{9.4}$$

with an obvious extension to other exponents. Total persistence was introduced by Cohen-Steiner et al. [105] in the context of stability analysis. With a suitable normalization factor, it can be made into the *p-norm* of a persistence diagram [93].



(a) $\sigma_{\text{Global}} = 1.0$ (b) $\sigma_{\text{Global}} = 0.373$ (c) $\sigma_{\text{Global}} = 0.328$ (d) $\sigma_{\text{Global}} = 0.141$

Figure 9.3: Some example clusterings of the 'two circles' data. Commonly, the left-most clustering is considered to be the most suitable, while the right-most clustering is deemed the least suitable. The values under each clustering refer to our novel global quality measure.

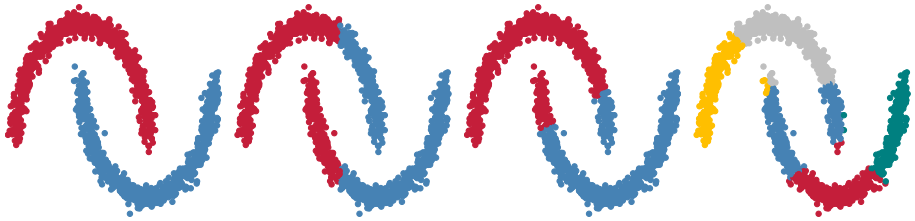
Conceptually, the total persistence of a function is similar to the concept of *total variation* [295] in mathematics. Like the total variation, $\text{Pers}(\cdot)$ measures the amount of changes that are characteristic of a function. It thus serves as a coarse characterization of function behaviour, both in terms of geometrical–topological variation and in terms of noise.

9.2.4 Assessing clusterings

In order to illustrate our method, we analyse the example depicted in Figure 9.3. What makes the clustering shown in Figure 9.3a more suitable than the clustering shown in Figure 9.3d? Here, we argue that the clustering is more suitable because it retains the *structural features*—the two circles—in the data, whereas these large-scale features are lost in the remaining clusterings. In order to assess whether structural features have been retained to some extent by a given clustering, persistent homology is the ideal tool due to its built-in notion of the scale of features and its stability under perturbations. We shall first take a look at how the calculation of persistent homology changes in the presence of a clustering. Formally, we consider a clustering $\mathcal{C} := \{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ to be a *partition* of the data.

Definition 9.3 (Partition). A *partition* of a set S is a set of non-empty subsets of S such that every element $s \in S$ occurs in exactly one of these subsets. In other words, S is the disjoint union of these subsets.

The previous definition only applies to 'hard clusterings', i.e. clusterings in which an object is assigned to exactly one cluster. We leave the treatment of other types of clusterings for future work. The partition generated by a clustering algorithm induces a partition of the auxiliary connectivity data structure used to calculate persistent homology. For example, if we use a Rips graph $\mathcal{R}_\epsilon$, the clustering induces a partition of its vertex indices. This partition is defined by connecting vertices u and v if $(u, v) \in \mathcal{R}_\epsilon$ and $(u, v) \in \mathcal{C}_i$ for some i. Hence, edges are only kept—with their edge weight unmodified—if both vertices are in the same



(a) $\sigma_{\text{Global}} = 1.0$ (b) $\sigma_{\text{Global}} = 0.069$ (c) $\sigma_{\text{Global}} = 0.104$ (d) $\sigma_{\text{Global}} = 0.034$

Figure 9.4: Some example clusterings of the 'two spirals' data. σ_{Global} decreases rapidly because of the simple geometry of the data.

cluster. The result is a set of Rips graphs, each corresponding to a cluster $\mathcal{C}_i \in \mathcal{C}$. In a similar manner, this partition also induces a partition of the Vietoris–Rips complex $\mathcal{V}_\epsilon$ of the data. This partition can be calculated efficiently and only requires calculating the Vietoris–Rips complex once for the unclustered data.

We may now calculate persistent homology on each of the partitioned data structures, resulting in a set of persistence diagrams

$$\mathcal{D}_{\mathcal{C}} := \{\mathcal{D}_1, \ldots, \mathcal{D}_k\}, \tag{9.5}$$

where each diagram $\mathcal{D}_i$ measures a subset of the geometrical–topological features of a function f present in the partition induced by $\mathcal{C}$. Returning to the example depicted in Figure 9.3, we define a clustering $\mathcal{C}$ to be suitable if most of the prominent features of f are preserved. Ideally, we would like the persistence diagram $\mathcal{D}_f$ of the original data to be the disjoint union of the individual diagrams $\mathcal{D}_i$, i.e.

$$\mathcal{D}_f = \mathcal{D}_1 \uplus \cdots \uplus \mathcal{D}_k, \tag{9.6}$$

with $\mathcal{D}_i \cap \mathcal{D}_j = \varnothing$ for $i \neq j$. In this case, each feature of f remains represented in exactly one unique cluster of $\mathcal{C}$. We can use *total persistence* to measure the amount of features that are lost. If Equation 9.6 holds, we have

$$\operatorname{Pers} \mathcal{D}_f = \sum_{i=1}^{k} \operatorname{Pers} \mathcal{D}_i, \tag{9.7}$$

so we are able to reconstruct the geometrical–topological variation of f perfectly. This definition does not necessarily require that clusters are perfectly separable. It is sufficient for the prominent features of the data descriptor function to be preserved. In case of the eccentricity descriptor, for example, this means that a clustering must not split up parts of the data set that are considered central.

Global assessment of a clustering

Practically, Equation 9.7 almost never holds. But we may measure how much a given clustering deviates from it by calculating

$$\sigma_{\text{Global}} := \frac{\sum_{i=1}^{k} \operatorname{Pers}(\mathcal{D}_i)}{\operatorname{Pers}(\mathcal{D}_f)}, \tag{9.8}$$

i.e. the fraction of total persistence that is retained by the clustering. σ_{Global} has a range of $[0, 1]$, with 1 meaning that the amount of geometrical–topological variation has been fully retained by the clustering, and 0 meaning that all features have been lost.

Examples

Figure 9.3 and Figure 9.4 depict some example clusterings and their corresponding σ_{Global} values. Both data sets are well-known in the data mining community [289] as they depict interesting behaviour already in two dimensions. For these simple examples, the global quality measure appears to penalize clusterings with a large number of clusters—this is only due to the simple shape of the data here, though. Especially the second data set in Figure 9.4 does not exhibit any salient topological features other than the good separation of the two clusters. As a consequence, other clusterings are unable to preserve most of the features of the shape descriptor function.

A set of experiments that the author performed and subsequently published as *supporting information* to the original paper [314] demonstrates further stability properties of our measure. Among others, we showed that σ_{Global} does not decrease when multiple, but equally valid, clusterings of a set of linked circles are investigated.

Local assessment of a clustering

Assessing a clustering locally, i.e. on the level of individual clusters, is slightly more complex. To the best of our knowledge, there currently exists no stable measure that is capable of such an assessment without requiring class labels. The *silhouette coefficient* [320], for example, can be calculated for individual clusters, but it suffers from instabilities because it only focuses on distances. For our topology-based assessment, we first calculate the persistence diagram

$$\mathcal{D}_{f,i} := \mathcal{D}_f \cap \mathcal{D}_i, \tag{9.9}$$

which contains topological features of the original data that are retained in the i^{th} cluster. This does not account for topological features that are slightly changed by the partition because the connectivity of the partitioned Rips graph or the Vietoris–Rips complex changes as well.

For each unaccounted point p in $\mathcal{D}_i \setminus \mathcal{D}_{f,i}$ we find its nearest neighbour q, measured using the L_∞-distance, in $\mathcal{D}_f$. If

$$2\|p - q\|_\infty \leq \operatorname{pers}(p), \tag{9.10}$$

we match the two points and add q to $\mathcal{D}_{f,i}$. This accounts for a slight drift of points in a persistence diagram that has been previously observed by Kloke [222]. Equation 9.10 only accepts these points if the magnitude of the drift stays below their persistence. In case a point remains unmatched, this is the minimum amount by which it would increase the Wasserstein distance between the two diagrams. Hence, Equation 9.10 ensures that the errors made during this assignment never increase beyond the costs in the Wasserstein distance. The point q thus becomes the new representative for the topological feature described by p. We now calculate

$$\sigma_{\text{Local}} = \frac{\operatorname{Pers}(\mathcal{D}_{f,i})}{\operatorname{Pers}(\mathcal{D}_i)}, \tag{9.11}$$

i.e. the ratio of total persistence both present in the cluster and the complete data set to the amount of the total persistence in the cluster. σ_{Local} also has a range of $[0,1]$, with 1 meaning that all features found in the cluster are also present in the original data. The idea behind this measure is that a cluster should contain only features of the function f that are present in the original data set.

In summary, assessing a cluster on a local level permits a more nuanced analysis workflow. By considering clusters individually, even a partition that is unsuitable at a global level may contain some suitable clusters. We shall encounter such examples later on.

An intuitive view

Both global and local quality measures may be thought of as topological equivalents to the *explained variance* or *explained variation* measures from statistical modelling. In essence, we deem a clustering $\mathcal{C}$ to be suitable when it explains a large amount of the geometrical–topological variation present in the data. Using *total persistence* has several beneficial properties in comparison to methods that only assess the geometry of the data:

1. As a topological measure, it works for even very complex cluster shapes. Of course, this assumes that the selected data descriptor function is sufficiently discriminative.

2. Persistent homology is aware that structures in data may occur at multiple scales. Hence, it is unbiased with respect to the geometrical extents of clusters. A tight cluster may still contain multiple large-scale features and thus contribute a large amount of geometrical–topological variation.

3. At the same time, it is robust against noise because it considers the scale of features. $\mathrm{Pers}(\cdot)$ barely changes if only few low-persistence points in a persistence diagram are changed. This is especially relevant for clustering algorithms whose underlying models are of a stochastic nature. In essence, the stability properties of $\mathrm{Pers}(\cdot)$ imply that changing the cluster associations for a few points does not negatively impact the assessment. As we will see in Section 9.2.5, existing measures tend to exhibit instabilities.

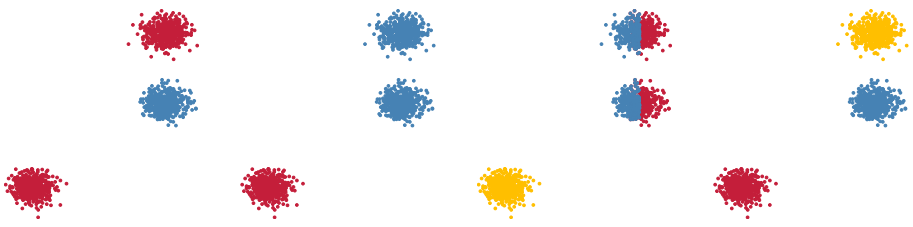
4. It permits the assessment of clusterings on the local level of individual clusters without requiring label information. This permits us to find suitable clusters in clusterings that are globally unsuitable. We will return to this aspect in Section 9.3.

Despite these advantages, our measure of course has some limitations. We will discuss them now briefly by means of several simple data sets.

Limitations

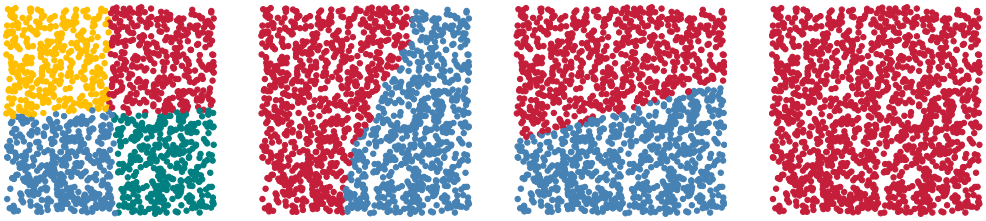
We first observe that our measure is incapable of assessing the similarity of different clusterings. This is already evident from the data set shown in Figure 9.4, for example. The absolute difference between the σ_{Global} values of Figure 9.4b and Figure 9.4c is larger than the absolute difference between Figure 9.4c and Figure 9.4d, even though Figure 9.4b and Figure 9.4c bear a closer resemblance to each other.

Likewise, our measure is unable to distinguish between clusterings where parts of some clusters are disconnected on large scales. Figure 9.5 depicts an example data set. To distinguish better between the individual 'blobs', they would have to be connected on extremely large scales. While this can always be achieved manually, the heuristics we introduced in Chapter 5, Section 5.4, p. 96 ff., eschew these connections in favour of the small-scale structures. Labelling points in different 'blobs' with the same cluster label neither creates nor destroys any new topological information, so our measure cannot detect any changes. However, as soon as a



(a) $\sigma_{\mathrm{Global}} = 1.0$ (b) $\sigma_{\mathrm{Global}} = 1.0$ (c) $\sigma_{\mathrm{Global}} = 0.66$ (d) $\sigma_{\mathrm{Global}} = 1.0$

Figure 9.5: Some example clusterings of the 'Gaussians' data. This data set has a very simple geometry and the individual clusters are well-separated. If separated 'blobs' are considered to be in the same cluster, our measure does not detect any changes in quality.



(a) $\sigma_{\text{Global}} = 0.64$ (b) $\sigma_{\text{Global}} = 0.64$ (c) $\sigma_{\text{Global}} = 0.66$ (d) $\sigma_{\text{Global}} = 1.0$

Figure 9.6: Some example clusterings of the 'Uniform distribution' data. We argue that only the last clustering is 'true' to the structure in the data. The splits introduced in the remaining clusterings are essentially only caused by statistical fluctuations.

clustering changes topological information, such as the one depicted in Figure 9.5c, it will be penalized by our measure. We do not consider this restriction to negatively affect the performance, as real-world data usually do not contain well-separated structures on large scales. Even if they did, clusterings such as the one shown in Figure 9.5a are not very likely.

Moreover, the topological assessment of clusterings fails if no salient features are present in the clusters. Figure 9.6 shows some example clusterings of a uniform distribution in $\mathbb{R}^2$. This is a classical example of data that does not exhibit any *clustering tendency* [402, pp. 457–461]. We shall later see that all existing clustering validity indices are incapable of detecting this. In practice, checking to what extent data may be clustered requires the usage of auxiliary statistics such as the Γ-index introduced by Hubert and Schultz [206] or the Cox–Lewis statistic [118]. While our measure does not attempt to replace these statistics, it is interesting to know that it is less prone to suggest a clustering tendency where none exists. Furthermore, since $\sigma_{\text{Global}} \in [0, 1]$, users can easily detect that something is amiss with the data—even for only two clusters, more than 40% of the geometrical–topological variation of the data is lost.

In addition to this evaluation on synthetic data, we will also evaluate the performance of our measure on real-world data. When analysing the 'Iris flower' data in Section 9.3.1, we will see that our measure is the only measure capable of suggesting both $k = 2$ and $k = 3$ as suitable choices for the number of clusters. While our experiments and our evaluation indicate that our novel topology-based measure performs well, a large-scale analysis on more data sets may be of interest for future work. At present, the author considers this to be outside the scope of this thesis, though.

9.2.5 Comparison with existing clustering validity indices

The introduction already alluded to certain issues with existing clustering validity indices. In the following, we briefly introduce common indices, evaluate their performance, and comment on their advantages as well as on their shortcomings. We assume that we are given

a data set $\mathbb{X}$ of cardinality n with a distance measure $\mathrm{dist}(\cdot,\cdot)$, which is usually a metric in the mathematical sense. This permits us to define the $n \times n$ distance matrix D with

$$d_{ij} := \mathrm{dist}(x_i, x_j), \tag{9.12}$$

which contains all pairwise distances between the input data points. Furthermore, we assume that we have a clustering $\mathcal{C}$ with k clusters, i.e. $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$, where each cluster $\mathcal{C}_i$ contains $n_i := \operatorname{card} \mathcal{C}_i$ points. Given two subsets $\mathbb{U} \subseteq \mathbb{X}$ and $\mathbb{V} \subseteq \mathbb{X}$, we define their distance as

$$\mathrm{dist}(\mathbb{U}, \mathbb{V}) := \sum_{x \in \mathbb{U}} \sum_{y \in \mathbb{V}} d_{ij}, \tag{9.13}$$

i.e. the sum of all distances between data points in the respective sets. This distance is always well-defined, regardless of whether the two subsets overlap. We are interested in two specific sets of distances that characterize the *cohesion* of a clustering.

Definition 9.4 (Intra-cluster distance). The *intra-cluster distance* is obtained by summing all distances between pairs of points in the same cluster, i.e.

$$\mathrm{dist}_{\mathrm{intra}} := \frac{1}{2} \sum_{i=1}^{k} \mathrm{dist}(\mathcal{C}_i, \mathcal{C}_i), \tag{9.14}$$

where we need the division by two because we must not count every pair of distances twice. Smaller values for $\mathrm{dist}_{\mathrm{intra}}$ are desirable because they indicate that clusters are compact.

Definition 9.5 (Inter-cluster distance). Letting $\overline{\mathcal{C}_i}$ denote the set-theoretic complement of a cluster, we can define the *inter-cluster distance* as

$$\mathrm{dist}_{\mathrm{inter}} := \frac{1}{2} \sum_{i=1}^{k} \mathrm{dist}(\mathcal{C}_i, \overline{\mathcal{C}_i}), \tag{9.15}$$

where we need a division by two for the same reason as above. Higher values for $\mathrm{dist}_{\mathrm{inter}}$ are desirable because they indicate that individual clusters are separated on a large scale.

Prior to using intra-cluster and inter-cluster distances to derive numerous clustering validity indices, we briefly discuss a construction that originates in graph theory. If we assume that the distance matrix D is symmetrical and has a diagonal of zero, which is always the case if the distance measure is a metric in the mathematical sense, we may consider it to be a *weighted*

adjacency matrix [25, p. 5 ff.] of the complete graph over n vertices. With this viewpoint, we may count the number of *intra-cluster edges* N_{intra} as

$$N_{\text{intra}} := \frac{1}{2} \sum_{i=1}^{k} n_i (n_i - 1), \tag{9.16}$$

and the number of *inter-cluster edges* N_{inter} as

$$N_{\text{inter}} = \sum_{i=1}^{k} \sum_{j=i+1}^{k} n_i n_j, \tag{9.17}$$

respectively. Both N_{intra} and the N_{inter} do not necessarily correlate with the quality of a clustering. They can merely indicate whether there are many small clusters, as opposed to a few big clusters.



With these definitions, we can derive some *clustering validity indices*. The subsequent indices are known as *internal* quality measures because they do not make use of information such as class labels—which we assume to be unavailable for most real-world applications. For an in-depth discussion and survey of multiple internal quality measures, we refer to Zaki and Meira [402, Chapter 17] or Xiong and Li [396].

BetaCV

The BetaCV measure calculates the ratio between the mean intra-cluster distance to the mean inter-cluster distance, i.e.

$$\text{BetaCV} := \frac{\text{dist}_{\text{intra}} / N_{\text{intra}}}{\text{dist}_{\text{inter}} / N_{\text{inter}}} = \frac{N_{\text{inter}}\ \text{dist}_{\text{intra}}}{N_{\text{intra}}\ \text{dist}_{\text{inter}}}, \tag{9.18}$$

where small values are considered to be better because they indicate that, on average, intra-cluster distances are smaller than inter-cluster distances. This implies that the clusters are well-separated, as opposed to overlapping.

C-index

The C-index relates the intra-cluster distance to the sum of certain extremal distances in the distance matrix. We have

$$\text{C-index} := \frac{\text{dist}_{\text{intra}} - \text{dist}_{\text{min}}(N_{\text{intra}})}{\text{dist}_{\text{max}}(N_{\text{intra}}) - \text{dist}_{\text{min}}(N_{\text{intra}})}, \tag{9.19}$$

where $\text{dist}_{\text{intra}}$ is again the sum of all intra-cluster distances, $\text{dist}_{\text{min}}(N_{\text{intra}})$ is the sum of the N_{intra} smallest distances in the distance matrix D—not including the diagonal—and $\text{dist}_{\text{max}}(N_{\text{intra}})$ is the sum of the N_{intra} largest distances. The C-INDEX has values in $[0,1]$. Smaller values are considered to be better because they indicate compact clusters.

WITHIN-CLUSTER-SCATTER

The *within-cluster-scatter* (WCS) is another name for the intra-cluster distance $\text{dist}_{\text{intra}}$ that we already encountered in Definition 9.4. Small values are considered good because they indicate 'tight' clusters. Some clustering algorithms, e.g. k-means [211], attempt to minimize this measure.

DUNN INDEX

The *Dunn index* (DI) measures the ratio between the minimum distance between points from different clusters and the maximum distance between points from the same cluster [139]. We obtain it by calculating

$$\text{DI} := \frac{\text{dist}_{\text{inter}}^{\text{min}}}{\text{dist}_{\text{intra}}^{\text{max}}}, \tag{9.20}$$

where we refer to the quantities

$$\text{dist}_{\text{inter}}^{\text{min}} := \min_{i \neq j} \{\text{dist}(x, y) \mid x \in \mathcal{C}_i, y \in \mathcal{C}_j\} \tag{9.21}$$

$$\text{dist}_{\text{intra}}^{\text{max}} := \max_{i} \{\text{dist}(x, y) \mid x, y \in \mathcal{C}_i\} \tag{9.22}$$

as the *minimum inter-cluster distance* and the *maximum intra-cluster distance*, respectively. A large Dunn index corresponds to a good clustering because it indicates that even the closest distance between points in different clusters is larger than the maximum distance within a cluster. Hence, the Dunn index is maximized when we have very compact clusters that are extremely far from each other.

NORMALIZED CUT MEASURE

The *normalized cut* (NC) measure is motivated by graph theory. If we take a single cluster $\mathcal{C}_i$ from the clustering $\mathcal{C}$, the distances of all edges with at least one point in the cluster is an indicator of the volume of the $\mathcal{C}_i$. We denote this sum of distances by $\text{dist}(\mathcal{C}_i, \mathbb{X})$. If we consider $\mathcal{C}_i$ to induce a cut in the graph, the weight of the cut is given by all edges that go outside the cluster $\mathcal{C}_i$. Hence, $\mathcal{C}_i$ induces a cut whose weight is $\text{dist}(\mathcal{C}_i, \overline{\mathcal{C}_i})$. The normalized

cut index now measures the total sum of the ratio between the weight of the cut and the volume of the cluster, i.e. the sum of all its edge weights. Formally, we have

$$\mathrm{NC} := \sum_{i=1}^{k} \frac{\mathrm{dist}(\mathcal{C}_i, \overline{\mathcal{C}_i})}{\mathrm{dist}(\mathcal{C}_i, \mathbb{X})}, \tag{9.23}$$

where higher values indicate better clusterings because they imply that the inter-cluster edges have larger distances than the intra-cluster edges. Thus, this measure also considers small intra-cluster distances to be indicative of a good clustering.

Silhouette coefficient

The *silhouette coefficient* measures both the separation of clusters as well as their internal connectivity. We first calculate a *local silhouette coefficient* s_x as

$$\mathrm{s}_x := \frac{b_x - a_x}{\max\{a_x, b_x\}}, \tag{9.24}$$

where a_x is the average distance of point x to all other points within its cluster, and b_x is the average of all distances of points x to points in the closest other cluster. We have $\mathrm{s}_x \in [-1, +1]$, where +1 shows that x is much closer to points in its own cluster and removed from other clusters, 0 indicates that x is on a cluster boundary, and −1 indicates that x is closer to another cluster than its own—which may indicate that x has been assigned to the wrong cluster. Finally, the silhouette coefficient $\mathrm{s}_{\mathcal{C}}$ of a clustering $\mathcal{C}$ is defined as the mean value of s_x across all points. While the silhouette coefficient can be calculated for individual clusters, it is commonly used as a graphical aid in cluster analysis only [320].

Expressive power

The indices sometimes lack expressive power and are outperformed by our topological measure. We calculated the different indices on four different data sets to demonstrate this. Table 9.1 shows a comparison between the different indices for different data sets. The best value of a measure is typeset in boldface. Empty cells in the table indicate undefined values. Except for the third data set, our measure exhibits no limitations in detecting suitable clusterings.

The Dunn index appears to agree with the assessment of our measure very often. However, a detailed examination shows that it is very prone to instabilities. For example, when we slightly perturb the clustering shown in Figure 9.3a, the Dunn index drops from 0.064 to 0.012 even though less than 0.5% of the data points have been assigned to different clusters. With this value, the Dunn index leads us to consider the clustering shown in Figure 9.3b to be the most suitable one. For the same perturbation, our measure σ_{Global} exhibits an absolute change that

Clustering	BetaCV	C-index	WCS	DI	NC	s_C	σ_{Global}
Figure 9.3a	0.90	0.44	1283.5	**0.064**	**2.39**	−0.09	**1.0**
Figure 9.3b	**0.0**	0.28	1170.4	0.018	1.35	0.27	0.373
Figure 9.3c	**0.0**	0.22	1052.1	0.006	1.52	**0.35**	0.328
Figure 9.3d	0.31	**0.08**	**620.0**	0.009	4.79	**0.35**	0.141
Figure 9.4a	0.56	0.17	957.4	**0.157**	1.56	0.389	**1.0**
Figure 9.4b	**0.0**	0.18	966.3	0.005	1.56	0.388	0.069
Figure 9.4c	0.47	0.09	856.0	0.011	1.62	**0.499**	0.104
Figure 9.4d	0.26	**0.06**	**500.0**	0.008	**4.80**	0.456	0.034
Figure 9.5a	**0.0**	0.446	259.9	0.08	1.48	0.266	**1.0**
Figure 9.5b	**0.0**	0.002	127.8	**0.92**	1.79	0.801	**1.0**
Figure 9.5c	0.15	0.124	507.6	0.0	2.77	0.280	0.66
Figure 9.5d	0.11	**0.001**	**61.8**	0.29	**2.91**	**0.804**	**1.0**
Figure 9.6a	0.42	**0.08**	**669.6**	**0.013**	**3.74**	**0.413**	0.64
Figure 9.6b	**0.0**	0.22	1052.5	0.002	1.54	0.346	0.64
Figure 9.6c	**0.0**	0.21	1045.6	0.005	1.52	0.353	0.66
Figure 9.6d		1.0	1354.9		0.0		**1.0**

Table 9.1: Clustering validity indices for different data sets. Empty cells indicate undefined values. We can see that the Dunn index is often capable of detecting a good clustering. However, it turns out to be highly unstable when changing the class assignments of even a small number of points. For the third clustering, we can see that our measure considers markedly different clusterings to be of the same quality. For the last clustering, however, we argue that only the clustering with a single large cluster is true to the structure of the data. The other indices attempt to detect artificial structural features here.

is of the order of 10^{-4}, which makes the difference essentially negligible. In particular, σ_{Global} still considers the clustering to be more suitable than any other clustering of the same data set. Similar issues occur for the other clustering validity indices. The silhouette coefficient s_C, for example, increases from -0.09 to 0.11 for the perturbed clustering. Somewhat paradoxically, the added noise thus *improves* the assessment of the clustering. For other clusterings, the effects can be even more pronounced. If we perturb the clustering in Figure 9.4a in a similar manner, the Dunn index decreases from 0.157 to 0.0042, which amounts to only 3% of its previous value. Again, the changes in σ_{Global} are negligible.

9.2.6 Visualization methods

We provide two visualization methods that offer different views of the data. First, given multiple clusterings of a data set, we group them using the *clustering similarity graph*. The graph permits users to assess the complexity of the problem. If many clusterings appear to agree, for example, the data may contain a simple structure. Second, we enable the comparison and assessment of individual clusters of a data set through the *cluster map*. The map helps understand the patterns underlying a given cluster and makes it possible to assess whether they are interesting and informative.

In the following, we will colour-code clusterings (using the values of σ_{Global}) and individual clusters (using the values of σ_{Local}). Since both measures have the same range, we can use the same colours to indicate the amount of explained topological variation in the data. Blue ● indicates values in $[0.80, 1.00]$, yellow ● indicates values in $[0.60, 0.80)$, and red ● indicates values less than 0.60. These ranges have been inspired by statistical modelling where being able to account for less than 60% of the variation is usually taken to be the sign of a bad model.

Clustering similarity graph

To handle the global comparison of multiple clusterings, we first require a similarity measure. As the comparison of different clusterings is a vital ingredient of any data analysis process, numerous similarity measures already exist [167, 205, 264, 303, 373]. We prefer similarity measures that are also metrics in the mathematical sense because we want to compare multiple partitions among each other. Re-iterating the short discussion from Chapter 8 about metrics, we require the *triangle inequality* to be satisfied, i.e. $\text{dist}(x, y) \leq \text{dist}(x, z) + \text{dist}(z, y)$, for three clusterings x, y, and z. The triangle inequality ensures that two clusterings x and y that are similar to the same cluster z must by necessity be similar to each other as well. Most similarity measures do not satisfy this inequality, which results in inconsistent similarity assessments. We thus opted for using the *Mirkin metric*.

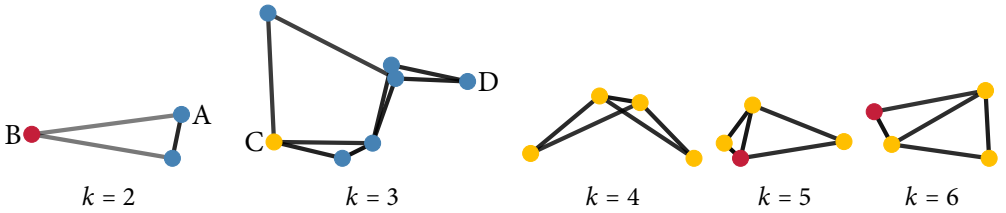


Figure 9.7: Clustering similarity graphs for the 'Iris flower' data. We can see that the global quality of different clusterings drops for $k = 4$. Here, no clustering is capable of preserving more than 80% of the geometrical–topological information in the data. The labels refer to several selected clusterings for the subsequent analysis.

Definition 9.6 (Mirkin metric). Let x and y be two clusterings of the same data with a cardinality of n. Furthermore, let n_{01} be the number of pairs of points that are in different clusters under x but in the same cluster under y. Let n_{10} be defined vice versa. The *Mirkin metric* between the clusterings is defined as

$$\mathrm{dist}_{\mathrm{Mirkin}}(x, y) := \frac{2(n_{01} + n_{10})}{n^2}, \tag{9.25}$$

which is a variation of the Rand index [303]. The factor in the denominator is a normalization factor that is due to Meilă [264], who observed that the Mirkin metric is not bounded with respect to the number of points.



The Mirkin metric is known to work less well when comparing clusterings with a different number of clusters among each other [264]. This poses no problem for our visualization because we only use the metric to compare clusterings with the same number of clusters. Having a metric on all clusterings, we build the *clustering similarity graph* visualization by showing each partition as a node. The Mirkin metric yields a matrix of pairwise distances between the nodes and we arrange them using force-directed graph visualization techniques [25, Chapter 10] to obtain an embedding in 2D. This works similar to the MDS algorithm for embedding a matrix of distances, although graph drawing algorithms usually aim to prevent certain configurations, such as overlapping nodes or edges. We only show the edges between the two nearest neighbours of a given partition in order to support the orientation of the user within the graph. To depict the overlap between two clusterings, we use the opacity of edges and modify it according to the *Rand index*, which is a common measure of the overlap between clusters. Hence, an edge that connects two dissimilar clusterings appears to be almost transparent, while similar clusterings are connected by darker edges.

Definition 9.7 (Rand index). Let x and y be two clusterings of the same data with a cardinality of n. Similar to the Mirkin metric, let n_{11} be the number of pairs of points that are in the same clusters under both x and y, and n_{00} be the number of pairs of points that are in different clusters under both x and y. The *Rand index* between x and y is defined as

$$s_{\text{Rand}}(x, y) := \frac{n_{11} + n_{00}}{\binom{n}{2}}, \tag{9.26}$$

where the normalization factor in the denominator is due to Hubert and Arabie [205] in order to remove the influence of the cardinality of the input data.

While the Rand index is not a proper metric, it has the advantage of being easy to understand. Two partitions with a large overlap should be placed nearby and also, presumably, explain the data similarly well. Node colours (as defined above) correspond to the values of σ_{Global} and indicate how well a clustering overall retains features in the data.

Example and usage Figure 9.7 shows several clustering similarity graphs of the 'Iris flower' data, which we will analyse in Section 9.3.1. We can use the graphs to figure out the most plausible number of clusters. It is interesting to note that clusterings for $k = 3$ are most expressive. Even though we calculated the same number of clusterings for every parameter k, many of them will be equal and lead to overlapping nodes in the graph. However, for $k = 3$, there is a large amount of variation, but also a high degree of similarity between 'neighbouring' clusterings, as indicated by the edge opacity. For a larger number of clusters, we observe that the values for σ_{Global} drop rapidly. Without knowing more about the data, we may thus consider either $k = 2$ or $k = 3$ to be a suitable number of clusters. Later on, we will see that this intuition turns out to be correct.

Cluster map

To permit the exploration of individual clusters in a data set, we provide a combination of a glyph-based view and a simplified density projection of the data. We first use a dimensionality reduction algorithm such as *principal component analysis* (PCA) to obtain a two-dimensional embedding of the original data set. This embedding will serve as an invariant map of the data. Instead of using only the calculated coordinates of the embedding, we calculate a hexagonally-binned plot to visualize the density distribution of the data. We colour each cell according to the number of points it contains using a standard greyscale colour map in which darker colours indicate more points. Hexagonally-binned plots are known to have better data aggregation properties than rectangularly-binned plots [73, 74, 75]. Furthermore, the visualization of density is known to support reasoning about different partitions [356].

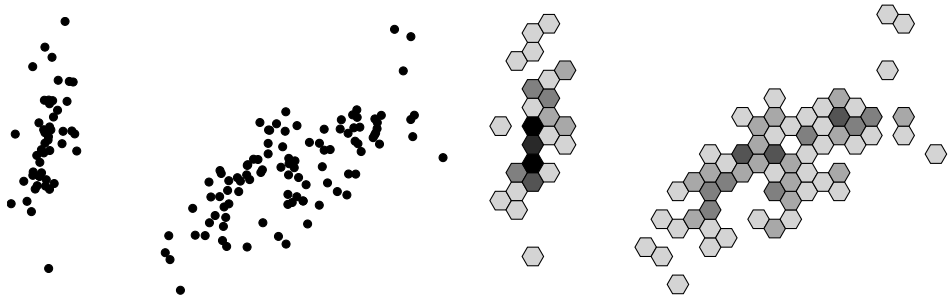


Figure 9.8: An embedding and its hexagonal density plot. Without having to resort to any 'formal' density estimation methods, the hexagonal binning results in a useful visualization of density changes. For this example, we used the 'Iris flower' data.

Figure 9.8 shows the advantages of the hexagonal density plot for multivariate data. The plot furthermore permits us to depict additional information—such as cluster boundaries—on the data. We will make use of this feature in Section 9.3.2, for example, where we analyse different clusterings of the 'Olive oils' data.

A second component of our visualization, in addition to the hexagonal density plot, is a set of glyphs for representing individual clusters. Each glyph consists of a modified *star plot* [86] that depicts a simplified representation of the data points within the cluster. This form of visualization has turned out to be an informative representation of multivariate data [157]. Here, we do not represent individual data points but use a band that shows the minimum, mean, median, and maximum value of each attribute—i.e. each dimension—for all data points in the cluster. Mean and median have been selected to indicate whether the distribution of values of a specific dimension within a given cluster is skewed. Together with the visualization of the minimum and maximum values, this improves understanding the 'profile' of a cluster [157]. The background of each glyph is colour-coded according to the respective value of σ_{Local}. It indicates how well a cluster matches the geometrical–topological features present in the data descriptor function f.

The glyphs are placed automatically along the map to minimize clutter. Each glyph is then connected to the centroid—the geometric centre—of the cluster it represents in order to highlight cluster placements. Furthermore, our cluster glyphs use the concept of *semantic zooming* [202] to offer more details on demand. When zooming in on a single cluster glyph, the star plot is furnished with additional labels—corresponding to the attributes present in the data—and more lines are progressively shown. This reduces the amount of visual clutter. The cluster glyphs also provide standard interaction techniques. As outlined above, they may be used to trigger the visualization of cluster boundaries, for instance.



Figure 9.9: Cluster glyphs of a perfect clustering. The captions refer to the different species in the 'Iris flower' data. We can see that the individual species are characterized by different cluster profiles. Please refer to Table 9.2 for the attribute names.

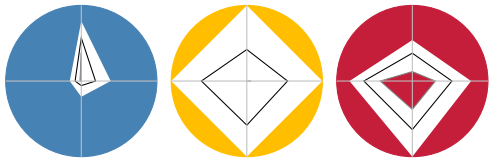


Figure 9.10: Cluster glyphs of a bad clustering. The background colours of the glyphs indicate that only one cluster preserves geometrical–topological features to a sufficiently high extent.

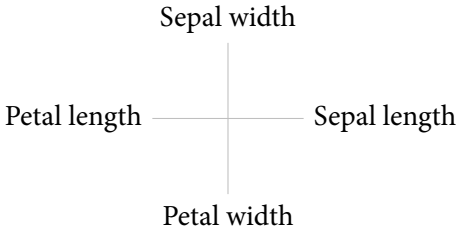
Example and usage Figure 9.9 and Figure 9.10 show two sets of cluster glyphs for different clusterings of the 'Iris flower' data, which we will subsequently analyse. Without any prior knowledge about the data, we can see that the first clustering is more plausible than the other clustering. This is indicated by the distinct profile of one of the cluster glyphs, which contains flowers with extremely large sepal widths and extremely small petal lengths. The background colours of the glyphs help assess individual clusters. For the second clustering, we can see that the first cluster preserves the geometrical–topological features of the shape descriptor function, whereas the remaining clusters are incapable of preserving them. This shows that the clustering algorithm has problems with separating individual clusters, which in turn hints at complex cluster geometries. Nonetheless, at least one partition appears to be usable. In a real-world analysis scenario, users could now for example remove the 'good' cluster and focus on separating the two 'bad' clusters. Such information is not available by existing measures.

9.3 Results

In the subsequent sections, we shall briefly demonstrate our technique by analysing different clusterings of multivariate data sets of varying complexities. We purposefully selected data sets that are known to be challenging to cluster in order to highlight the benefits of our approach. All the clusterings were obtained using clustering algorithms from the scikit-learn toolkit [289]. Whenever possible, we applied automated parameter tuning. Since the goal of this chapter is not the evaluation of different clustering algorithms but rather their

Attribute	Minimum	Maximum
Sepal length	4.3	7.9
Sepal width	2.0	4.4
Petal length	1.0	6.9
Petal width	0.1	2.5

A short summary of all variables



Attribute names in the cluster glyph

Table 9.2: Attributes in the 'Iris flower' data. The data are 'well-behaved' because all attributes are continuous, have the same unit (cm), and even similar orders of magnitude. For layout reasons, we do not repeat individual attribute names in every figure.

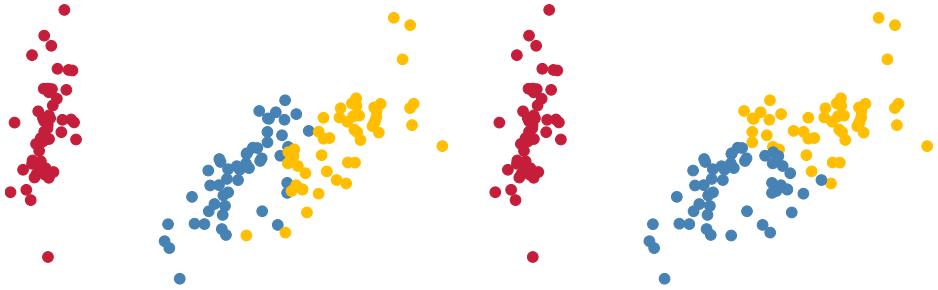
clusterings, we deliberately refrain from describing every clustering result—and especially every clustering algorithm—in detail. Instead, we exemplarily explain interesting properties of selected clusterings by means of our visualization techniques and our quality measures.

9.3.1 'Iris flower' data

The 'Iris flower' data set is well known in data mining. It was first recorded by Anderson [6] and has since become a standard data set for multivariate data analysis. The data set contains 150 measurements of 4 attributes of 3 different 'Iris' flower species—*I. setosa*, *I. versicolor*, and *I. virginica*. See Table 9.2 for a short description of the attributes. We obtained the data from the UCI Machine Learning Repository [247], which contains a corrected and cleaned variant of the data set. Even though the individual measurements of the flowers are well-behaved (they are all continuous, have the same unit, and similar orders of magnitude), the data set is challenging because the flowers cannot be clustered correctly without knowing the species information. There are two well-defined clusters in the data, one containing *I. setosa*, the other one containing the remaining two species. Splitting the second cluster is not easy because its boundaries are unclear. Figure 9.11 on p. 250 illustrates this by showing the 'ground truth' species assignments along with another clustering. Even though we used an optimized variant of the k-means algorithm [328], the second cluster is split incorrectly.

Comparison with existing clustering validity indices

Different clustering validity indices also consider the 'Iris flower' data to be a challenging data set. Table 9.3 on p. 250 shows their values along with the values of our new global measure σ_{Global}. Most measures pick up on the pronounced decomposition into $k = 2$ clusters in the data. No existing measure is capable of detecting that $k = 3$ is also a valid number of clusters. Our topological measure σ_{Global} proves that $k = 3$ is still a viable option, with less



Ground truth k-means clustering with $k = 3$

Figure 9.11: Example clusterings of the 'Iris flower' data. We can see that the species assignments are somewhat contrary to the cluster boundaries, leading many algorithms to create clusters that contain both *I. versicolor* and *I. virginica*.

than 4% of the geometrical–topological variation in the data being lost. Additionally, it is the only measure capable of indicating that clusterings with $k \geq 4$ are significantly less suitable than clusterings for $k = 2$ or $k = 3$. We may reach the same result by referring to the clustering similarity graphs. They permit us to observe the behaviour of different clustering algorithms without knowing the correct species assignments. Figure 9.7 on p. 245 shows the clustering similarity graphs of selected clusterings with different amounts of clusters. We can see that starting with $k = 4$, most clusterings are incapable of retaining important features in the data. Without using class labels or clustering validity indices, our topological measure thus suggests that $k = 2$ clusters and $k = 3$ clusters are more suitable than $k \geq 4$ clusters. In the following, we will refer to the individual clusters as depicted in Figure 9.12.

	BetaCV	C-index	WCS	Dunn index	NC	s_C	σ_{Global}
$k = 2$	**0.0**	**0.056**	89.90	**0.339**	1.652	**0.630**	**1.0**
$k = 3$	0.187	0.069	90.60	0.098	2.763	0.480	0.967
$k = 4$	0.215	0.059	81.76	0.105	3.792	0.434	0.627
$k = 5$	0.253	0.062	72.67	0.117	4.812	0.353	0.561
$k = 6$	0.235	0.057	**67.74**	0.133	**5.820**	0.383	0.561

Table 9.3: Clustering validity indices for the 'Iris flower' data. For every k, we have used the best possible clustering with respect to the species labels, measured using the Rand index. The optimal value for every measure is highlighted. We observe that our measure is the only quality measure that exhibits a sharp drop between $k = 3$ and $k = 4$, which indicates that at this point, important features are being destroyed by clusterings.

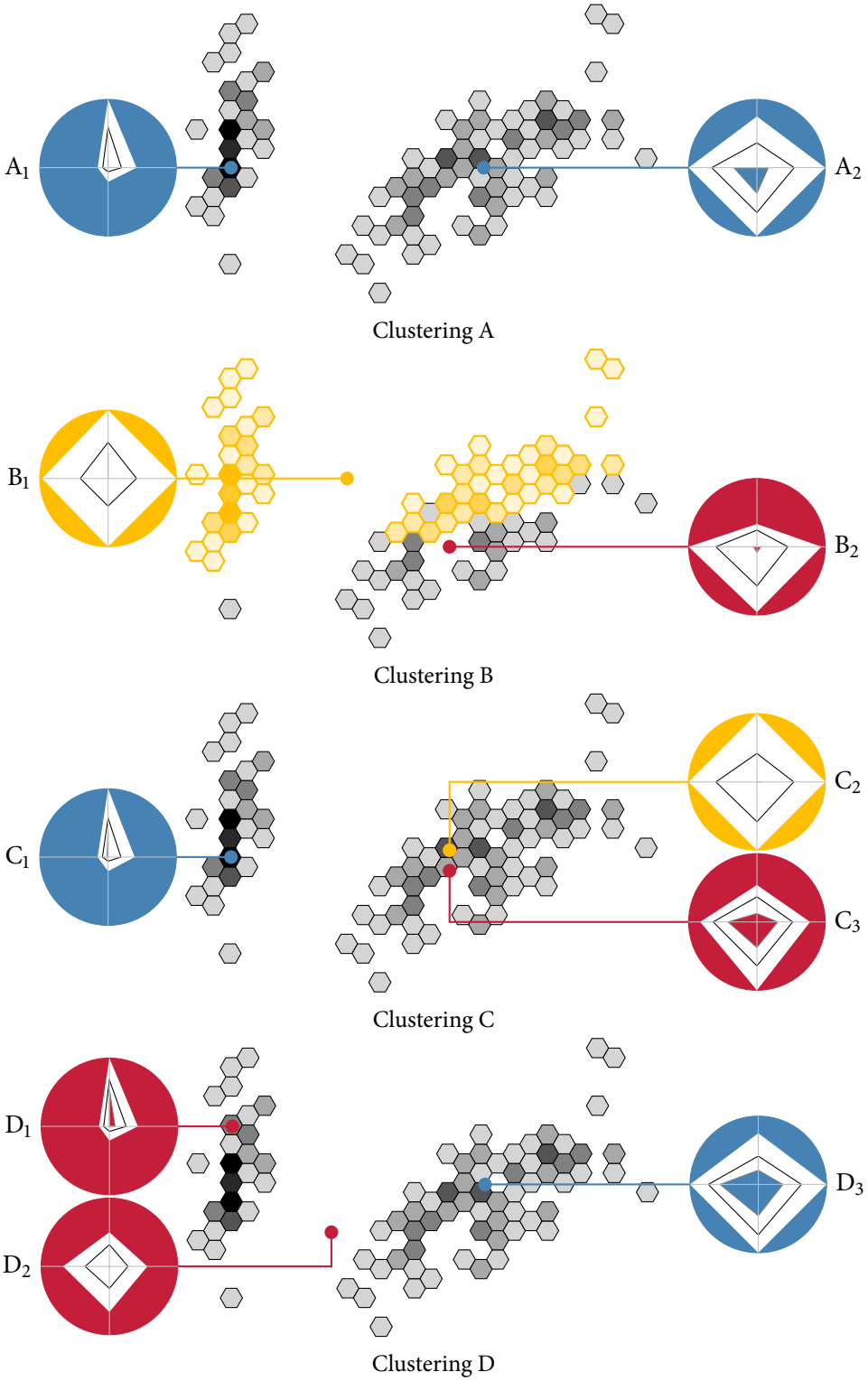


Figure 9.12: Cluster maps for the 'Iris flower' data. We selected a set of diverse clusterings with interesting clusters to discuss the advantages of our visualization. For B_1, we exemplarily depict the cluster extents.

Two clusters

We first analyse two clusterings with $k = 2$ in order to obtain an intuition for the different visualizations. Clustering A contains the correct species assignments. We can see that both A_1 and A_2 retain all features of the data. In fact, they satisfy $\sigma_{\text{Local}} = 1.0$. This is not too surprising, though, because any clustering algorithm is capable of splitting these data into two different clusters. The placement of the cluster centroids suggests that both clusters have simple shapes. The corresponding glyph shows that flowers in A_1 have small petal lengths, small petal widths, and extremely large sepal widths. By contrast, A_2 contains flowers with significantly larger petal lengths and petal widths.

Clustering B appears to be very different. It is the only clustering with a low σ_{Global} value, indicated by its node colour ● in the clustering similarity graph—see Figure 9.7 on p. 245. As a consequence, it is far removed from the other clusterings in the graph. The cluster map shows that B_1 and B_2 also have low σ_{Local} values, making these clusters dubious. The centroid placement and the cluster extents of B_1 confirm this. We coloured every cell in the background of the cluster map that is associated with B_1, and we observe that the cluster is not connected. Furthermore, the extremal bands shown in the glyph for B_1 indicate that this cluster also contains flowers with small sepal widths, just like B_2 does. This clustering is thus far from optimal, just as indicated by our measures.

Three clusters

We next analyse clusterings with $k = 3$. The spatial proximity of many clusterings and high edge opacities in the clustering similarity graph indicate a strong overlap between most of the partitions. All clusterings are assigned values of $\sigma_{\text{Global}} \geq 0.90$, except one, which turns out to be of medium quality ● in the clustering similarity graph. We refer to this clustering as C and compare it with one of the remaining clusterings.

At first glance, the cluster glyphs seem to indicate that C_1 and D_1 are very similar. A closer look shows that the centroid is placed slightly differently, because D_1 'misses' several of the *I. setosa* flowers, in particular those with small values in the first two attributes. Since this cluster is very pronounced and already exists at small scales of the Rips graph $\mathcal{R}_\epsilon$, breaking it partially up results in a lower σ_{Local} value in comparison to C_1. This is again a demonstration of how our measure σ_{Local}, coupled with visualization techniques, helps assess the clusters on a local level. Both C_1 and D_3 are locally suitable in the sense that they represent a subset of the data correctly. In an interactive analysis scenario, analysts could now remove the corresponding measurements and attempt to find better clusters on the remaining part of the data, for example.

More clusters

The clustering similarity graph also helps evaluate the behaviour of clusterings with a larger number of clusters. Figure 9.7 on p. 245, shows what happens when we increase the values of k. Already for $k = 4$, all clusterings have $\sigma_{\text{Local}} < 0.80$, meaning that less than 80% of the geometrical–topological features are preserved globally by the clustering. For $k = 5$ and $k = 6$, a few clusterings remain stable with respect to σ_{Global} because they prefer splitting up *I. versicolor* and *I. virginica* prior to splitting up the more compact and concise *I. setosa* cluster. This helps retain some features of the data.

We also observe that clusterings become progressively dissimilar, as indicated by edges with higher transparency, because there are more possibilities for partitioning the data points. This demonstrates how the clustering similarity graph, in combination with σ_{Global}, can be used to quickly explore the overall suitability of different partitions without having to explore them on a local level.

9.3.2 'Olive oils' data

As a second data set, we use the 'Olive oils' data. It is commonly included in data analysis packages [112]. The data set contains the ratios of 8 different fatty acids (palmitic, palmitoleic, stearic, oleic, linoleic, arachidic, linolenic and eicosenoic) for 572 olive oils, produced in 9 different regions (North Apulia, South Apulia, Calabria, Sicily, Inland Sardinia, Coastal Sardinia, Umbria, East Liguria, and West Liguria). A common data analysis question is whether it is possible to distinguish olive oils from different regions just by their fatty acid compositions [18]. The task is made more difficult by the fact that there are at least two valid clusterings. The first one uses three clusters to separate the oils into coarse regions, namely North, South, and Sardinia. The second one uses a more fine-grained decomposition into the original nine regions.

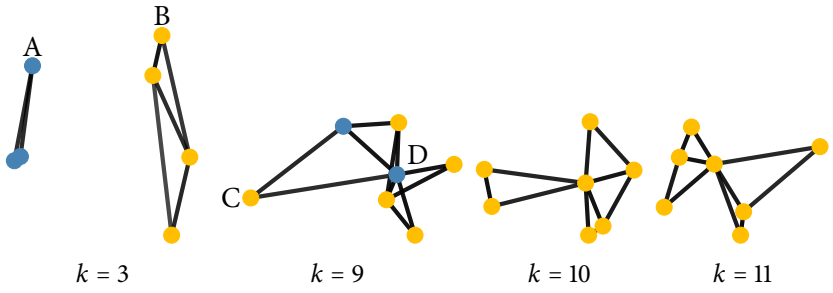
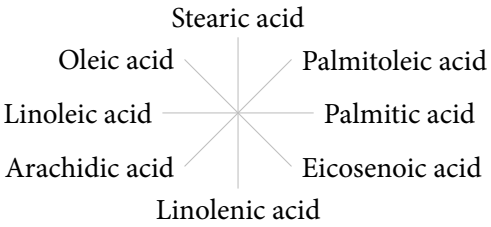


Figure 9.13: Clustering similarity graphs for the 'Olive oils' data. As soon as the 'natural' number of clusters in the data has been surpassed, the global quality of the clustering drops significantly. The labels refer to several selected clusterings for the subsequent analysis.

Acid	Median	Mean
Palmitic	52.50	55.27
Palmitoleic	35.80	41.92
Stearic	31.80	34.47
Oleic	47.50	47.95
Linoleic	56.90	52.11
Arachidic	43.10	41.66
Linolenic	57.30	54.55
Eicosenoic	26.80	25.53

A short summary of all variables



Attribute names in the cluster glyph

Table 9.4: Attributes in the 'Olive oils' data. The concentration of every acid varies between 0 and 100, so we only report their median and mean value. For layout reasons, we do not repeat the names of the individual acids in every figure.

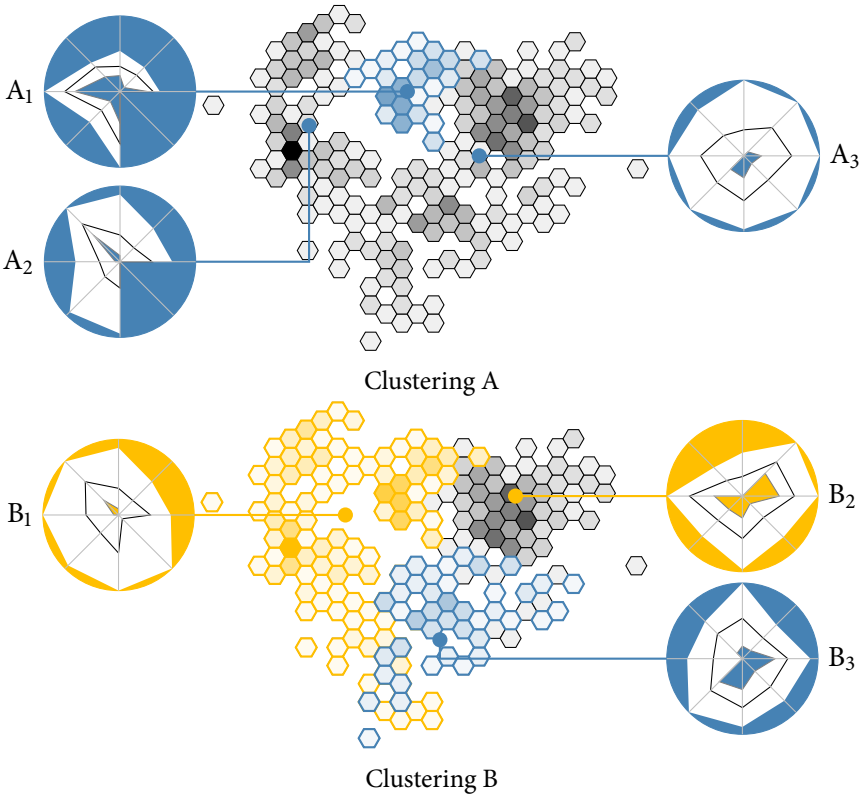


Figure 9.14: Cluster maps for the 'Olive oils' data and $k = 3$. The centroid placement of B_1 is usually indicative of a problematic clustering. Furthermore, this cluster overlaps with B_3, as we can see from the visualized cluster extents.

In the following, we will take a look at clusterings with $k = 3$, $k = 9$, and more clusters. We do not discuss clusterings with $k \in [4, 8]$ because they do not contain any interesting clusters or 'failure' cases. Figure 9.13 shows several clustering similarity graphs for the data. Again, without knowing any class labels, we can see that the value of σ_{Global} start to decay after about $k = 9$ clusters, meaning that clusterings only retain 60%–80% of the geometrical–topological features. The individual clusterings are very similar, as indicated by the high edge opacities. See Table 9.4 for information about the attributes as well as their names in the cluster glyphs.

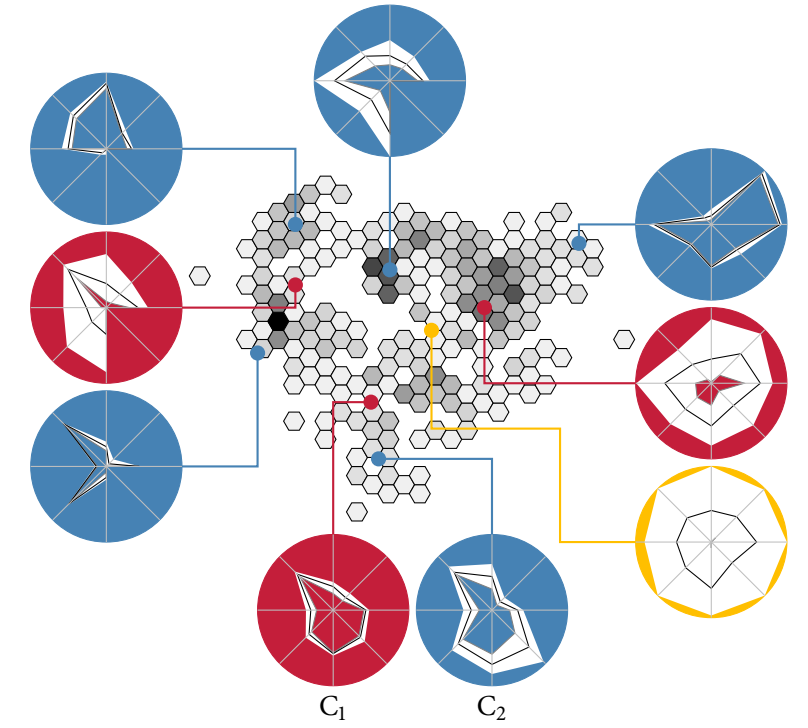
Three clusters

We first compare clusterings A and B with each other. Figure 9.14 shows the corresponding cluster maps. Clustering A is a perfect partition into the three different 'macro-regions' mentioned above. It retains all features in the data perfectly, i.e. $\sigma_{\text{Global}} = \sigma_{\text{Local}} = 1.0$. The cluster glyphs show that A_1 and A_2 contain oils without any eicosenoic acid. The bands also show how A_1 differs from A_2. Oils in A_1 have e.g. lower amounts of oleic acid and higher amounts of linoleic acid than oils in A_2. A_3 is mostly characterized by non-zero amounts of eicosenoic acid. The cluster extents of A_1—coloured according to its quality—show that it is a small cluster in comparison to the other clusters. A_2 is smaller than A_3. Few overlaps occur and the boundaries are placed in sparse areas, whereas centroids are placed near dense areas—see A_2, for example.

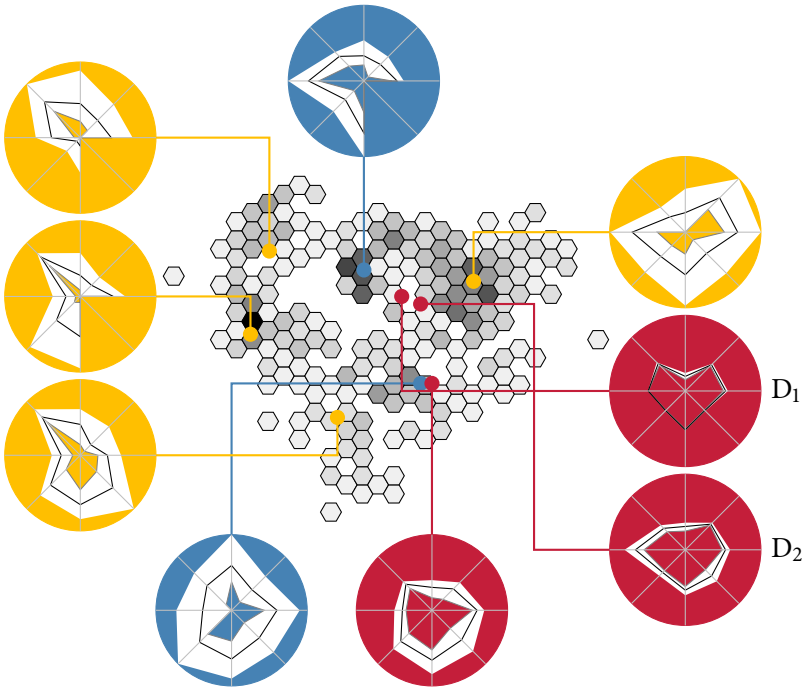
Clustering B exhibits lower σ_{Local} values. B_1, for example, does not fully capture features in the data because it contains some oils with high amounts of eicosenoic acid but otherwise similar fatty acid compositions compared to oils in A_2. The cluster extents show some of the cells as being disconnected from the remaining cells in the cluster. This assignment destroys the simple shape of the cluster, leading to a lower σ_{Local} value. B_3 consists of oils with, on average, higher amounts of palmitic and palmitoleic acids. The cluster glyphs indicate that some overlaps with oils in B_2 exist, which explains the lower σ_{Local} value. Overall, this clustering is still very informative because we learn about a different set of non-obvious subgroups. This shows the benefits of using both σ_{Global} and σ_{Local} to assess clusterings.

Nine clusters

The clustering similarity graph for $k = 9$ shows that all clusterings satisfy $\sigma_{\text{Global}} \geq 0.60$. As indicated by their distances and edge opacities in the graph, the clusterings are rather similar, except for a single outlying one. Due the amount of clusters, we cannot compare all clusters of all clusterings with each other, so we keep the subsequent analysis short. Figure 9.15 on p. 256 shows the cluster maps. Cluster C_1 contains only few oils and shares similar characteristics to oils in cluster C_2, whose centroid is located nearby in the map. The split between C_1 and C_2



Clustering C



Clustering D

Figure 9.15: Cluster maps for the 'Olive oils' data and $k = 9$. Using the cluster glyphs, we can easily see that one cluster is the same in both clusterings.

thus seems arbitrary and is penalized by our measure σ_{Local} because these oils are connected on all scales in the Rips graph $\mathcal{R}_\epsilon$ due to their similar composition. This clustering was obtained using DBSCAN [160]; a slight perturbation of its parameters results in merging these clusters, which shows that the split was not justified in the first place.

Clustering D has the best σ_{Global} value of 0.951. Here, we observe two new clusters, D_1 and D_2, that do not appear anywhere else. Their oils are characterized by average amounts of all acids and a slightly above-average amount of linoleic acid. However, the oils in D_1 are too similar to some oils in D_2, leading our measure to consider this as a problematic cluster. This is also expressed in the proximity of the cluster centroids. Interestingly, neither D_1 nor D_2 are consistent with respect to the original classes of the data—their oils come from all three different regions in Italy. We consider this split to be nonetheless 'interesting' in the sense that it shows a hitherto-unknown partition in the data. This analysis shows how our geometrical–topological assessment helps detect informative clusterings that go beyond class label information. Moreover, we have seen that our workflow is even able to support the detection and correction of instabilities in clustering algorithms.

9.3.3 'El Niño' data

In Chapter 6, Section 6.4.2, p. 137 ff., we already encountered the El Niño phenomenon. To briefly re-iterate the earlier discussion, we recall that the El Niño phenomenon refers to a powerful pattern in world climate that is characterized by a distinct anomaly in sea surface temperatures in the Pacific Ocean. The formation of El Niño is still not fully understood, but it is known that the phenomenon causes catastrophic weather in many parts of the earth. It occurs at irregular intervals (3–7 years) and may last up to 2 years. In the following, we will analyse the 'El Niño' data set from the UCI Machine Learning Repository [247]. The data set contains 178,080 buoy measurements of 5 continuous attributes in the Pacific Ocean, comprising a period of 18 years. The complex parameter space and its size make this data set

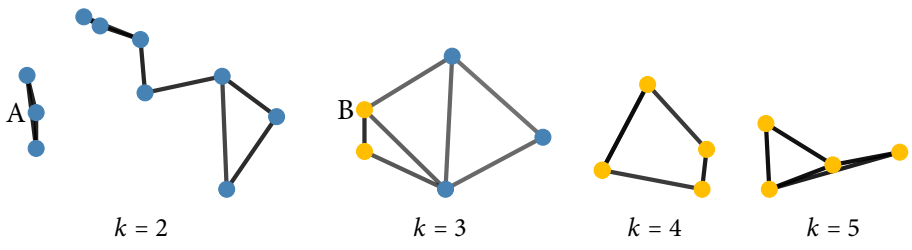
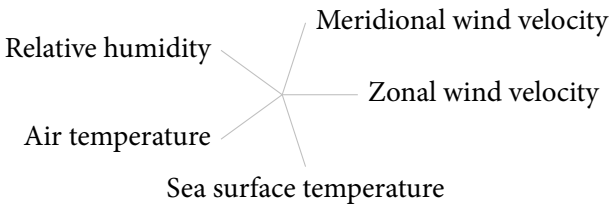


Figure 9.16: Clustering similarity graphs for the 'El Niño' data. Due to the large size of the data set, we obtained less 'useful' clusterings for $k \geq 4$. After pruning away clusters that only contained a few data points, the output did not significantly differ from the results for $k = 2$. The labels refer to the two clusterings that appear in the subsequent analysis.

Attribute	Unit
Buoy number	—
Date	—
Latitude	°
Longitude	°
Zonal wind velocity	$\mathrm{m\,s^{-1}}$
Meridional wind velocity	$\mathrm{m\,s^{-1}}$
Relative humidity	%
Air temperature	°C
Sea surface temperature	°C

A short summary of all variables



Attribute names in the cluster glyph

Table 9.5: Attributes in the 'El Niño' data. We only use the last five attributes in the subsequent analysis because they are continuous. Including spatio-temporal information would make the analysis even more complex. For layout reasons, we do not repeat attribute names in every figure.

very challenging to cluster [112, Chapter 3]. Figure 9.16 depicts different clustering similarity graphs for the 'El Niño' data, while Table 9.5 shows a short description of all attributes. All clustering algorithms that we employed exhibited problems when handling the data. These are caused by the large number of measurements, which often differ only by small amounts. We observed decreases in both σ_{Local} and σ_{Global} values already for $k \geq 3$. These could conceivably have been avoided by different pre-processing techniques, but we are more interested in finding out what topological analysis may contribute in this case.

Two clusters

We exemplarily discuss a clustering with two clusters (A) first. With $\sigma_{\text{Global}} \approx 0.978$, it retains more than 97% of the geometrical-topological features of the data. Figure 9.17 shows the corresponding cluster map. The data set shows up as a high-density core with density decreasing towards the boundary. This is a typical behaviour of more complex multivariate data sets that has already been observed in the literature [214]. Instead of visualizing the cluster extents, we only highlight the overlaps between the two partitions. We can see that the overlaps are restricted to a small area. The cluster glyphs show that A_2 contains measurements with—on average—much warmer air temperatures and sea surface temperatures than the ones in A_1. Their σ_{Local} values are high, so we may consider both clusters to be 'trustworthy'. Since El Niño is commonly associated with extraordinary warm sea surface temperatures, this clustering is very informative. Referring back to the data set, we found that measurements from A_2 indeed predominantly arose in El Niño years.

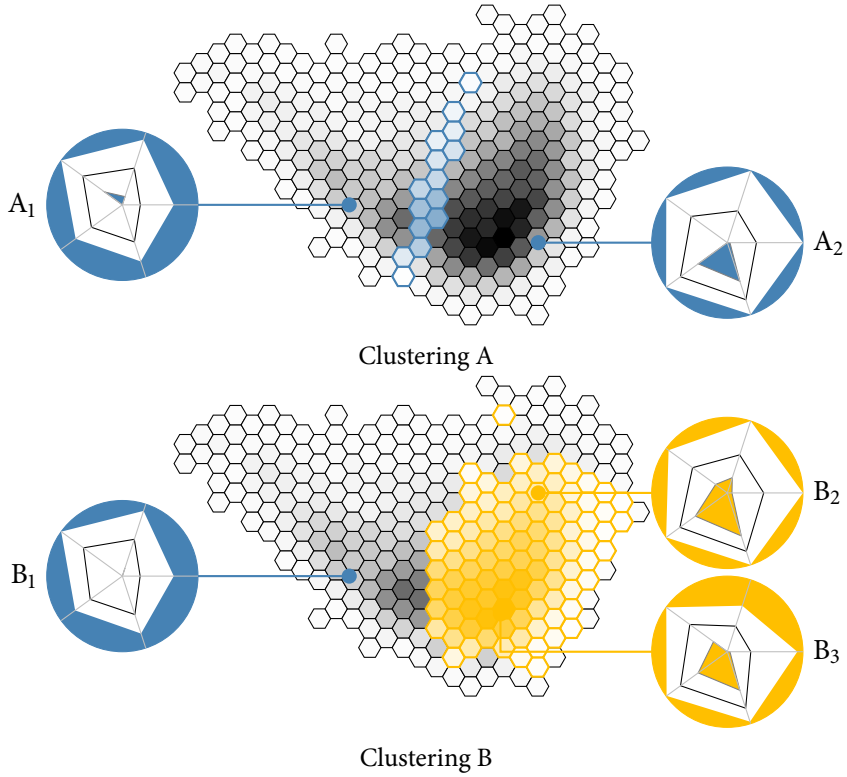


Figure 9.17: Cluster maps for the 'El Niño' data. The split of cluster A_2 into B_2 and B_3 is penalized in the second clustering because it arbitrarily separates very similar measurements in the attribute space, leading to the creation of clusters that do not correspond to a natural split.

It is interesting to see that the structural changes that we detected in Chapter 6, Section 6.4.2, p. 137 ff., within a similar data set by means of the *simplicial chain graph* also show up—less obviously—in a standard cluster analysis workflow. This demonstrates that anomalies such as El Niño give rise to large-scale changes in the internal structure of data. Our topology-based measures are capable of detecting these changes and assess them accordingly.

Three clusters

Next, we exemplarily discuss one clustering with three clusters (B). It has $\sigma_{\text{Global}} \approx 0.77$, hence almost 80% of the features of the data are retained. In contrast to A, clustering B exhibits more overlaps in cluster extents. In particular, the extents of B_2 and B_3 have a very large overlap—we visualize it in Figure 9.17. The glyph colours show that B_2 and B_3 have $\sigma_{\text{Local}} < 0.80$. Each cluster turns out to lose about 25% of the geometrical–topological variation. This loss can be explained when looking at the value distributions in the glyph bands. Neither mean values nor spread fully explain why data points have been assigned to one cluster instead of the other.

The clear distinction between 'extraordinary measurements' and 'regular measurements' as present in clustering A is not apparent here. This clustering also shows the advantages of assessing clusters individually. The glyph for B_1 indicates that it retains at least 80% of the features in the data. In fact, B_1 has $\sigma_{\text{Local}} \approx 0.999$, meaning that this cluster fits the global structure of the data extremely well.

Summary

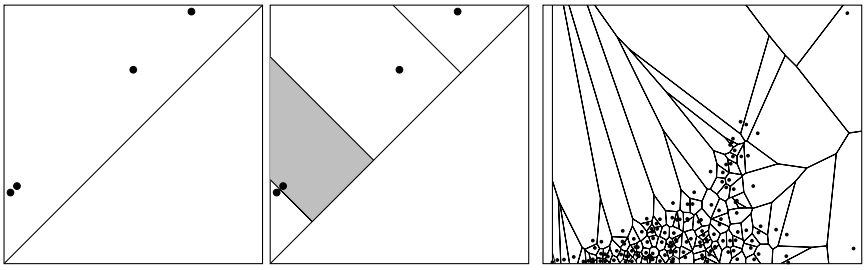
In summary, even though clustering algorithms found this data set challenging, their results still reveal useful information about patterns in the data. Our visualization, combined with the values for σ_{Global} and σ_{Local} guides our attention and ensures that we do not have to treat clustering results as 'black boxes'.

9.4 Discussion

We presented two visualization techniques for supporting users in exploring and comparing different clusterings of multivariate data sets. Globally, our *clustering similarity graphs* permit the rapid exploration of different clusterings by arranging them using a clustering similarity measure. Locally, our *cluster maps* create a shared reference coordinate system coupled with glyphs for representing individual clusters that supports the comparison of clusters among each other and among different clusterings. Our methods are driven by two novel measures based on persistent homology that assess the geometrical–topological properties of a clustering and individual clusters. We demonstrated the utility of our visualization techniques by analysing three data sets of varying complexities.

Limitations

Neither our visualizations nor our measures are costly to realize, but they do not scale to substantially more than approximately ten dimensions. The individual cluster glyphs become rather unwieldy for more dimensions, which could be resolved by e.g. hierarchical displays. Similarly, the cluster map only generates uncrowded layouts until about ten clusters. After that, a *details-on-demand* [335] approach should be used. Nonetheless, these limitations still leave a lot of leeway for interesting data sets. For higher-dimensional data sets, care needs to be taken when selecting a dimensionality reduction method for generating the cluster map. An evaluation algorithm that integrates well into our framework was recently proposed by the author of this thesis [315].



A simple example A more complex example

Figure 9.18: Combining information theory and persistence diagrams. The left-hand side shows a synthetic persistence diagram whose entropy would be maximal according to the measure by Chintakunta et al. [101]; when calculating *spatial entropy* based on Voronoi cells, however, the diagram is judged differently. The right-hand side depicts Voronoi cells for a more complex example. The diagram has been transformed in order to simplify calculations. The author considers a measure based on the relative area of the Voronoi cells to be indicative of the spatial entropy of the persistence diagram.

Future work

Future work could expand the system to handle *fuzzy clusterings*, i.e. clusterings with underlying probability distributions [139]. Furthermore, our global and local quality measures could conceivably be decoupled from the visualization and used in other contexts. For instance, there are numerous popular *decision tree* algorithms [394] such as C4.5 [301], which could employ a criterion based on persistent homology in order to decide where to split data for classification. It would be interesting to see the results of this synthesis between clustering, classification, and persistent homology.

There are also several potential improvements for the visualizations. The individual cluster glyphs, for example, could be extended to show all data points as semi-transparent bands. Similar to *continuous scatterplots* [20], this would communicate more information about the shape of a cluster to the user. Likewise, different strategies for arranging the cluster map could be explored, such as *self-organizing maps* [223] or more abstract displays that focus only on a selected set of data points and their relations. For finding these representative data points, a recent method by Joia et al. [218], based on the SVD of a projection matrix, may be useful.

As for the exploitation of more topological features, we currently do not make use of any metrics between persistence diagrams, such as the ones discussed in Chapter 4, Section 4.6.1, p. 68 ff. This is partially because they were never meant to be used in the context of matching a smaller *part* of a data set against a larger data set. It is conceivable that a modification of the *bottleneck distance* [141, pp. 180–185] may lead to more precise results. The author suspects that this requires a relaxation of the metric definition, maybe to the extent that one merely uses a 'one-sided quasi-metric'.

An alternative to persistence diagram metrics is given by methods from *information theory*, which help quantify topological information. In the context of visualization, different approaches already use information theory [97, 215, 382], but its potential for TDA has been ignored so far. It would be interesting to quantify changes in topological activity, especially those changes that are induced by different clustering algorithms. In essence, a suitable clustering should preserve the entropy of the original data. A quantification of topological entropy would also be beneficial for time-varying data and ensemble data. In the opinion of the author, such *topological entropy* measures should incorporate the spatial characteristics of persistence diagrams. Previous work by Chintakunta et al. [101] only concentrates on discrete attributes of a persistence barcode. A mathematically more solid approach needs to make use of spatial properties of persistence diagrams. For these questions, geographical approaches, such as the one pioneered by Batty [26, 27], are helpful. The author is convinced that Voronoi diagrams [268], due to their stability with respect to small perturbations [305], could be used to provide a suitable decomposition of the domain of the persistence diagram into cells. A notion of spatial entropy could then be derived from the areas of these cells. Figure 9.18 on p. 261 illustrates this proposition. Yet another viewpoint is given by the theory of point processes, where early work by McFadden [263] results in a well-defined notion of entropy.

10 Conclusion

This chapter briefly summarizes the contributions of this thesis, discusses their implications, and gives an outlook to possible future work. Compared to the discussions in the individual chapters, the subsequent text takes a more panoramic view and points out novel opportunities for research that arise from this thesis.



The first part of this thesis focused on *qualitative aspects* of high-dimensional data. The goal was to provide discriminative visualizations of structural features. Chapter 5 introduced one such visualization technique, the persistence rings, which provide an improved visualization of the persistence tuples that occur when calculating persistent homology. In contrast to the state-of-the-art visualizations in computational topology, persistence rings provide a compact overview that does not suffer from overplotting. Hence, they permit a better perception of the multi-scale behaviour of topology. In addition, the chapter provided a novel exploratory data analysis workflow that leverages topological information within a density-based clustering approach. We demonstrated the utility by analysing complex multivariate data sets containing curvature information. Furthermore, we showed that these data sets are not amenable to standard visualization techniques.

The persistence rings demonstrate that low-dimensional, information-rich visualizations of the persistence tuples are possible. In a similar manner, the persistence diagram itself can be furnished with additional geometric information. The theory of *point processes* [117] provides an interesting metaphor here. If the appearance of topological features in the persistence diagram is treated as a particular instance of an underlying random process, structures in persistence diagrams can be quantified in a mathematically solid way. Moreover, point processes—and their higher-order characteristics—give rise to additional patterns that can be used to summarize persistence diagrams. Likewise, the empty space in a persistence diagram could be used to display further information about the geometry of topological features, for example. The mapping of arbitrary objects to the space below the diagonal could be accomplished using techniques such as the *Schwarz–Christoffel mapping* [135].

In addition, the chapter demonstrated that there are many open research questions concerning connectivity estimation in multivariate data. Future research should hence also target the development of multi-scale neighbourhood graphs and the analysis of their topological properties, such as their homotopy type preservation.

In Chapter 6, we developed a novel description of features in persistent homology, using *localized simplicial chains*. In essence, these features integrate a large amount of geometrical information into the otherwise mostly combinatorial topological workflow. We derived a new measure of geometrical conciseness for topological features and presented a novel algorithm for computing persistent homology with the express goal of optimizing this measure. In contrast to previous work, the approach in this thesis makes full use of the geometrical information that is present in a simplicial complex. Moreover, we showed that it may be implemented more easily and scales better. We used the measure to build the *simplicial chain graph*, a new visualization that focuses on visualizing ensemble data or time-varying data. The simplicial chain graph depicts a mixture of connectivity information and geometrical extents of topological features in data sets. We demonstrated the efficacy of such an approach by analysing qualitative changes in the structure of high-dimensional point clouds from two application domains, political analysis and climate science. A question that naturally arises from this visualization technique is how to depict changes in topological structure over time. In the opinion of the author, this requires the development of novel graph visualization techniques as well as the development of novel summary statistics for topological changes. The interplay of these two fields is necessary for obtaining informative visualizations. Furthermore, Chapter 6 also hints at the potential benefits arising from an analysis of graphs and networks by topological means. Given suitable modifications, persistent homology is likely to lead to salient multi-scale descriptors of graph and network topology, as well.

The second part of this thesis focused on developing *quantitative visualizations*. Here, the goal was to provide quantifiable information about properties of multivariate data and represent this information in an accessible manner. The novel methods presented in this part are particularly useful to assess data under different aspects, for example with respect to their topological dissimilarity. In Chapter 7, we analysed embeddings of multivariate data sets under two different aspects. First, we used existing quality measures for dimensionality reduction algorithms. By making their values available locally at every point in the data set, we were able to treat them as a scalar field. We then introduced a new algorithm for comparing the topological features of these scalar fields in order to find out whether different quality measures agree with their assessment of an embedding. Consequently, we referred to this approach as *agreement analysis*. Our agreement analysis helped uncover issues with different embeddings, such as erroneously-depicted linear structures or incorrect neighbourhoods. For the second aspect, we introduced a number of functions—*data descriptors*—that are specifically geared

towards describing salient properties of multivariate data sets. We used these descriptors to quantify to what extent an embedding preserves certain important properties, such as density. This led us to the derivation of a novel workflow for assessing the suitability of embeddings of high-dimensional data. We analysed the robustness of this approach and demonstrated its efficacy for generic data analysis tasks.

Both approaches point to similar opportunities for future research, namely the systematic analysis and development of novel feature descriptors for multivariate data. In particular, other properties of high-dimensional manifolds—apart from the ones that were already treated in Chapter 7—need to be investigated as to whether they provide salient information and afford an effective calculation.

Following this, Chapter 8 provided two novel visual metaphors for depicting multivariate data. The first metaphor—*model landscapes*—demonstrated how a persistence-driven embedding of high-dimensional data helps in quantifying differences between complex regression analysis models. The model landscape was shown to outperform existing quality measures for these models with respect to its expressiveness. Furthermore, Chapter 8 extended the analysis of the previous chapter by developing *data descriptor landscapes*—a depiction of the behaviour of multiple data descriptors on multivariate data. We demonstrated how to use the data descriptor landscape to rapidly assess unstable and anomalous behaviour of dimensionality reduction algorithms.

The methods in this chapter show the potential of topology-driven embeddings. As they represent multivariate data that are subject to parameter variations, future work needs to provide these visualization methods with a notion of uncertainty concerning the input parameters. Such a notion of uncertainty requires the development of novel measures for assessing the topological stability. The author considers a fusion of methods from information theory and computational topology to be a highly-relevant area for future research.

Finally, Chapter 9 introduced a novel assessment method for clusterings of multivariate data. We derived two specialized measures based on extended persistent homology that permit the assessment of a given clustering under both global and local aspects. In comparison to existing clustering validity indices, our measures turn out to be more robust and highly-discriminative with respect to finding suitable clusterings. We demonstrated these beneficial properties on a variety of data sets. Moreover, the chapter presented two novel visualizations—the *clustering similarity graph* and the *cluster map*—that are driven by our new quality measures. We showed how these individual parts may be combined in order to assess the suitability of a given clustering, both with respect to the number of clusters and their individual composition.

A natural extension of our method involves the development of clustering algorithms that specifically aim to preserve topological features in multivariate data when defining individual clusters. In addition, this chapter also raises the question of effective descriptors for clusterings.

This again underlines the need for salient shape descriptors of multivariate data. Furthermore, the chapter showed that persistent homology requires novel metrics that permit a partial comparison between persistence diagrams.



This thesis was motivated by the large amount of multivariate data sets in many application domains, which require new visualization and analysis modalities. The main goal was to demonstrate how concepts from algebraic topology—most prominently *persistent homology*—can be used to visualize both quantitative and qualitative aspects of complex multivariate data sets. Throughout the thesis it became clear that topological methods are capable of augmenting, supporting, and even surpassing existing approaches for multivariate visual data analysis. A holistic understanding requires the cooperation of both geometrical and topological methods, though. This thesis thus serves as a stepping stone for increasing the acceptance of persistent homology for multivariate data analysis. A large amount of open questions, new research directions, and promising concepts lies ahead.

Acronyms

DI	Dunn index
DKRZ	German Climate Computing Centre
EDA	Exploratory data analysis
EMST	Euclidean minimum spanning tree
FA	Factor analysis
FLANN	Fast library for approximate nearest neighbours
HLLE	Hessian locally linear embedding
IPCC	Intergovernmental Panel on Climate Change
JCN	Joint contour net
KDE	Kernel density estimation
LLE	Locally linear embedding
LTSA	Local tangent space alignment
MDS	Multidimensional scaling
MRRE	Mean relative rank error
MSII	Multi-scale integral invariant
MST	Minimum spanning tree
NC	Normalized cut
PCA	Principal component analysis
PCP	Parallel coordinate plot
RMSE	Root-mean-square error
RP	Random projection
SNF	Smith normal form
SPE	Stochastic proximity embedding
SPLOM	Scatterplot matrix
SVD	Singular value decomposition
SVM	Support vector machines
t-SNE	t-distributed stochastic neighbour embedding
TAO	Tropical Atmosphere Ocean
TDA	Topological data analysis
WCS	Within-cluster-scatter

Glossary

$\mathcal{C}_\epsilon$	A Čech complex at maximum scale ϵ
$\mathcal{C}$	A clustering, consisting of individual clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$
$\mathcal{D}_f$	A persistence diagram of a scalar function f
ϵ	The scale parameter for persistent homology calculations
$\mathcal{L}_c(f)$	A level set of a scalar function f for a threshold c
$\mathcal{L}_c^+(f,c)$	A superlevel set of a scalar function f for a threshold c
$\mathcal{L}_c^-(f,c)$	A sublevel set of a scalar function f for a threshold c
L_f	Lipschitz constant of a Lipschitz-continuous function f
$\mathcal{M}$	A manifold
$\mathbb{R}_\infty$	The set of extended real numbers, $\mathbb{R} \cup \{\infty\}$
$\mathcal{R}_\epsilon$	A Rips graph at maximum scale ϵ
$\mathrm{s}_\mathcal{C}$	The silhouette coefficient of a clustering $\mathcal{C}$
$\mathrm{SO}(n)$	The special orthogonal group of $n \times n$ matrices
$\mathcal{V}_\epsilon$	A Vietoris–Rips complex at maximum scale ϵ

Index

Abstract simplex, *see* Simplex
Abstract simplicial complex, *see* Simplicial complex
Anscombe's quartet, 2

β-skeleton, 113
BetaCV measure, 240
Betti number
  persistent, 58
Betti numbers, 33
Bottleneck distance, 68
Bottleneck stability, 72
Boundary group, 32
Boundary homomorphism, 30

C-index, 240
Cascade, 61
Čech complex, 45
Chain complex, 32
Chain group, 29
  Relative, 34
Cluster map, 246
Clustering
  Persistence-based, 90
Clustering similarity graph, 244
Clustering validity indices
  BetaCV measure, 240
  C-index, 240
  Dunn index, 241
  Normalized cut measure, 241
  Silhouette coefficient, 242
  Within-cluster-scatter, 241
Complex
  Čech, 45
  Vietoris–Rips, 46
  Witness, 51
Cone
  of a simplicial complex, 231
Contour
  of a level set, 19
Contour tree, 20
Covering
  of a topological space, 42
Cuneiform, 100
Curvature, 102, 194
Cycle group, 32

Data descriptor, 170
Data descriptor landscape, 214
Density estimation, 91
Diameter, 46
Distance
  between persistence diagrams, 68
Dunn index, 241

El Niño, 137
Entropy
  spatial, 261

topological, 262
Essential homology class, 57
Extended persistence diagram, 232
Extended persistent homology, 229, 230

Filtration, 55
Extended, 231
Sublevel set, 63
Superlevel set, 63
Fundamental lemma
of simplicial homology, 31

Gabriel graph, 113
Geodesic ball, 120
Geodesic distance, 119
Geodesic distances, 123
Geometric graphs, 113
Gromov–Hausdorff stability, 75

Homeomorphism, 25
Homology class, 33
essential, 57
Homology group, 33
Persistent, 57
Relative, 35
Homotopy, 43
Homotopy equivalence, 44
Homotopy type, 44

Inter-cluster distance, 239
Intra-cluster distance, 239
Isometry, 75

Joint contour net (JCN), 22

k-skeleton, 28
of a simplicial complex, 28

Level set, 19
Lipschitz continuity, 73
Local continuity meta-criterion, 152
Localization algorithm, 118
Localization problem, 117
Lower star, 231

Manifold hypothesis, 18
MAPPER (algorithm), 22
Mean relative rank error, 153
Medoid, 129
Mirkin metric, 244
Model landscape, 202
Monotonic function, 55
Morse theory, 19
Morse–Smale complex, 21
MSII filter, 102
Multidimensional persistence, 223
Multidimensional size theory, 223
Multivariate data set, 5

Neighbourhood loss, 153
Nerve
of a covering, 42
Normalized cut measure, 241

p-norm, 144
Parallel coordinate plot (PCP), 15
Parameter selection
for Rips graphs, 96
Partition, 233
Persistence, 57
of a point in a persistence diagram, 65
Total, 232
Persistence barcode, 67
Persistence diagram, 65
Extended, 232
Persistence interval, 64
Persistence pair, 58
Persistence ring, 83

Layout algorithm, 85
Persistence tuple, 58
Persistence vineyards, 144
Persistence-based clustering, 90
Persistent Betti number, 58
Persistent homology
Calculation in arbitrary dimensions, 55
Calculation in dimension zero, 51
Performance improvements, 64
Stability, 72
Visualizations, 64
Persistent homology group, 57
Point processes, 263

Quality measures
Local continuity meta-criterion, 152
Mean relative rank error, 153
Neighbourhood loss, 153
Residual variance, 152
Root-mean-square error, 152
Spearman's rank correlation, 153
Stress, 152

Random walks, 113
Reeb graph, 20
Reeb space, 22
Residual variance, 152
Ricci curvature tensor, 195
Rips graph, 47, 113
Root-mean-square error, 152

Scatterplot matrix (SPLOM), 15
Silhouette coefficient, 242
Simplex, 27
Coface, 28
Face, 28
negative, 59
positive, 59
Simplicial chain, 30
Size, 121
Simplicial chain graph, 126, 127
Properties, 130
Simplicial chain group, *see* Chain group
Simplicial complex, 27
Vertices, 28
Simplicial homology, 27
Calculation, 35
Simplicial homology group, *see* Homology group
Size of a simplicial chain, 121
Smith normal form, 35
Spatial entropy, 261
Spearman's rank correlation, 153
Stability
of persistent homology, 72
Star
of a simplicial complex, 231
Star plot, 17
Stress, 152
Sublevel set, 19
Sublevel set filtration, 63
Superlevel set, 19
Superlevel set filtration, 63

Topological entropy, 262
Topological space, 26
Invariants, 26
Total persistence, 74, 232
Tropical Atmosphere Ocean data, 137

Upper star, 231

Vietoris–Rips complex, 46
Calculation, 47

Wasserstein distance, 69

Wasserstein stability, 74
Weight function, 50
Within-cluster-scatter, 241
Witness complex, 51

Bibliography

1. A. Adcock, D. Rubin, and G. Carlsson. 'Classification of hepatic lesions using the matching metric'. *Computer Vision and Image Understanding* 121, 2014, pp. 36–42. DOI: 10.1016/j.cviu.2013.10.014.

2. C. C. Aggarwal, A. Hinneburg, and D. A. Keim. 'On the surprising behavior of distance metrics in high dimensional space'. In: *Proceedings of the 8th International Conference on Database Theory*. Ed. by J. Van den Bussche and V. Vianu. Lecture Notes in Computer Science 1973. Springer, Heidelberg, Germany, 2001, pp. 420–434. DOI: 10.1007/3-540-44503-X_27.

3. C. C. Aggarwal and C. K. Reddy, eds. Data Mining and Knowledge Discovery. Chapman & Hall/CRC, Boca Raton, FL, USA, 2014.

4. D. K. Agrafiotis. 'Stochastic proximity embedding'. *Journal of Computational Chemistry* 24:10, 2003, pp. 1215–1221. DOI: 10.1002/jcc.10234.

5. M. Alexa and M. Wardetzky. 'Discrete Laplacians on general polygonal meshes'. *ACM Transactions on Graphics* 30:4, 2011, 102:1–102:10. DOI: 10.1145/2010324.1964997.

6. E. Anderson. 'The species problem in Iris'. *Annals of the Missouri Botanical Garden* 23:3, 1936, pp. 457–509. DOI: 10.2307/2394164.

7. T. W. Anderson. *An introduction to multivariate statistical analysis*. 3rd ed. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2003.

8. A. Andoni, D. Croitoru, and M. Pătraşcu. 'Hardness of nearest neighbor under L_∞'. In: *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*. Curran Associates, Inc., Red Hook, NY, USA, 2008, pp. 424–433. DOI: 10.1109/FOCS.2008.89.

9. M. Ankerst, S. Berchtold, and D. A. Keim. 'Similarity clustering of dimensions for an enhanced visualization of multidimensional data'. In: *IEEE Symposium on Information Visualization*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1998, pp. 52–60. DOI: 10.1109/INFVIS.1998.729559.

10. O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona. 'An extensive comparative study of cluster validity indices'. *Pattern Recognition* 46:1, 2013, pp. 243–256. DOI: 10.1016/j.patcog.2012.07.021.

11. M. Artin. *Algebra*. Prentice Hall, Upper Saddle River, NJ, USA, 1991.

12. S. Arya and H.-Y. A. Fu. 'Expected-case complexity of approximate nearest neighbor searching'. *SIAM Journal on Computing* 32:3, 2003, pp. 793–815. DOI: 10.1137/S0097539799366340.

13. D. Asimov. 'The Grand Tour: A tool for viewing multidimensional data'. *SIAM Journal on Scientific and Statistical Computing* 6:1, 1985, pp. 128–143. DOI: 10.1137/0906011.

14. D. Attali and A. Lieutier. 'Reconstructing shapes with guarantees by unions of convex sets'. In: *Proceedings of the 26th Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 2010, pp. 344–353. DOI: 10.1145/1810959.1811015.

15. D. Attali, A. Lieutier, and D. Salinas. 'Efficient data structure for representing and simplifying simplicial complexes in high dimensions'. *International Journal of Computational Geometry & Applications* 22:04, 2012, pp. 279–303. DOI: 10.1142/S0218195912600060.

16. D. Attali, A. Lieutier, and D. Salinas. 'Vietoris–Rips complexes also provide topologically correct reconstructions of sampled shapes'. *Computational Geometry* 46:4, 2013, pp. 448–465. DOI: 10.1016/j.comgeo.2012.02.009.

17. M. Aupetit. 'Visualizing distortions and recovering topology in continuous projection techniques'. *Neurocomputing* 70:7–9, 2007, pp. 1304–1330. DOI: 10.1016/j.neucom.2006.11.018.

18. A. Azzalini and N. Torelli. 'Clustering via nonparametric density estimation'. *Statistics and Computing* 17:1, 2007, pp. 71–80. DOI: 10.1007/s11222-006-9010-y.

19. B. Bach, C. Shi, N. Heulot, T. Madhyastha, T. Grabowski, and P. Dragicevic. 'Time curves: Folding time to visualize patterns of temporal evolution in data'. *IEEE Transactions on Visualization and Computer Graphics* 22:1, 2016, pp. 559–568. DOI: 10.1109/TVCG.2015.2467851.

20. S. Bachthaler and D. Weiskopf. 'Continuous scatterplots'. *IEEE Transactions on Visualization and Computer Graphics* 14:6, 2008, pp. 1428–1435. DOI: 10.1109/TVCG.2008.119.

21. M. Balasubramanian and E. L. Schwartz. 'The Isomap algorithm and topological stability'. *Science* 295:5552, 2002, p. 7. DOI: 10.1126/science.295.5552.7a.

22. M. Balzer and O. Deussen. 'Voronoi treemaps'. In: *IEEE Symposium on Information Visualization*. 2005, pp. 49–56. DOI: 10.1109/INFVIS.2005.1532128.

23. R. G. Baraniuk and M. B. Wakin. 'Random projections of smooth manifolds'. *Foundations of Computational Mathematics* 9:1, 2009, pp. 51–77. DOI: 10.1007/s10208-007-9011-z.

24. A. Bartkowiak and A. Szustalewicz. 'The Grand Tour as a method for detecting multivariate outliers'. *Machine Graphics & Vision* 6:4, 1997, pp. 487–505.

25. G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph drawing. Algorithms for the visualization of graphs*. Prentice Hall, Upper Saddle River, NJ, USA, 1999.

26. M. Batty. 'Entropy in spatial aggregation'. *Geographical Analysis* 8:1, 1976, pp. 1–21. DOI: 10.1111/j.1538-4632.1976.tb00525.x.

27. M. Batty. 'Spatial entropy'. *Geographical Analysis* 6:1, 1974, pp. 1–31. DOI: 10.1111/j.1538-4632.1974.tb01014.x.

28. U. Bauer, X. Ge, and Y. Wang. 'Measuring distance between Reeb graphs'. In: *Proceedings of the 30th Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 2014, pp. 464–473. DOI: 10.1145/2582112.2582169.

29. U. Bauer, M. Kerber, and J. Reininghaus. 'Clear and compress: Computing persistent homology in chunks'. In: *Topological Methods in Data Analysis and Visualization III*. Ed. by P.-T. Bremer, I. Hotz, V. Pascucci, and R. Peikert. Springer, Cham, Switzerland, 2014, pp. 103–117. DOI: 10.1007/978-3-319-04099-8_7.

30. U. Bauer, M. Kerber, and J. Reininghaus. 'Distributed computation of persistent homology'. In: *Proceedings of the 16th Workshop on Algorithm Engineering and Experiments* (ALENEX). Ed. by U. Meyer and C. C. McGeoch. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2014, pp. 31–38. DOI: 10.1137/1.9781611973198.4.

31. P. N. Belhumeur and D. J. Kriegman. 'What is the set of images of an object under all possible illumination conditions?' *International Journal of Computer Vision* 28:3, 1998, pp. 245–260. DOI: 10.1023/A:1008005721484.

32. M. Belkin and P. Niyogi. 'Towards a theoretical foundation for Laplacian-based manifold methods'. *Journal of Computer and System Sciences* 74:8, 2008, pp. 1289–1308. DOI: 10.1016/j.jcss.2007.08.006.

33. M. Belkin, J. Sun, and Y. Wang. 'Constructing Laplace operator from point clouds in $\mathbb{R}^d$'. In: *Proceedings of the 20th Annual ACM–SIAM Symposium on Discrete Algorithms*. Ed. by C. Mathieu. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2009, pp. 1031–1040. DOI: 10.1137/1.9781611973068.112.

34. P. Bendich and P. Bubenik. 'Stabilizing the output of persistent homology computations'. Preprint. 2015. URL: http://arxiv.org/abs/1512.01700.

35. P. Bendich, T. Galkovskyi, and J. Harer. 'Improving homology estimates with random walks'. *Inverse Problems* 27:12, 2011, p. 124002. DOI: 10.1088/0266-5611/27/12/124002.

36. J. L. Bentley. 'Multidimensional binary search trees used for associative searching'. *Communications of the ACM* 18:9, 1975, pp. 509–517. DOI: 10.1145/361002.361007.

37. M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational geometry. Algorithms and applications*. 3rd ed. Springer, Heidelberg, Germany, 2008. DOI: 10.1007/978-3-540-77974-2.

38. M. Berger. *A panoramic view of Riemannian geometry*. Springer, Heidelberg, Germany, 2003. DOI: 10.1007/978-3-642-18245-7.

39. S. Bergner, M. Sedlmair, T. Möller, S. N. Abdolyousefi, and A. Saad. 'ParaGlide: Interactive parameter space partitioning for computer simulations'. *IEEE Transactions on Visualization and Computer Graphics* 19:9, 2013, pp. 1499–1512. DOI: 10.1109/TVCG.2013.61.

40. P. Berkhin. 'A survey of clustering data mining techniques'. In: *Grouping multidimensional data. Recent advances in clustering*. Ed. by J. Kogan, C. Nicholas, and M. Teboulle. Springer, Heidelberg, Germany, 2006, pp. 25–71. DOI: 10.1007/3-540-28349-8_2.

41. M. Bernstein, V. de Silva, J. C. Langford, and J. B. Tenenbaum. 'Graph approximations to geodesics on embedded manifolds'. Manuscript. 2000. URL: http://isomap.stanford.edu/BdSLT.pdf.

42. S. Biasotti, L. De Floriani, B. Falcidieno, P. Frosini, D. Giorgi, C. Landi, L. Papaleo, and M. Spagnuolo. 'Describing shapes by geometrical–topological properties of real functions'. *ACM Computing Surveys* 40:4, 2008, pp. 1–87. DOI: 10.1145/1391729.1391731.

43. S. Biasotti, D. Giorgi, M. Spagnuolo, and B. Falcidieno. 'Reeb graphs for shape analysis and applications'. *Theoretical Computer Science* 392:1–3, 2008, pp. 5–22. DOI: 10.1016/j.tcs.2007.10.018.

44. S. Biasotti and S. Marini. '3D object comparison based on shape descriptors'. *International Journal of Computer Applications in Technology* 23:2–4, 2005, pp. 57–69. DOI: 10.1504/IJCAT.2005.006465.

45. S. Biasotti, S. Marini, M. Mortara, G. Patanè, M. Spagnuolo, and B. Falcidieno. '3D shape matching through topological structures'. In: *Discrete Geometry for Computer Imagery*. Ed. by I. Nyström, G. S. di Baja, and S. Svensson. Lecture Notes in Computer Science 2886. Springer, Heidelberg, Germany, 2003, pp. 194–203. DOI: 10.1007/978-3-540-39966-7_18.

46. G. Biau, F. Chazal, D. Cohen-Steiner, L. Devroye, and C. Rodríguez. 'A weighted k-nearest neighbor density estimate for geometric inference'. *Electronic Journal of Statistics* 5, 2011, pp. 204–237. DOI: 10.1214/11-EJS606.

47. E. Bingham and H. Mannila. 'Random projection in dimensionality reduction: Applications to image and text data'. In: *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD). ACM Press, New York, NY, USA, 2001, pp. 245–250. DOI: 10.1145/502512.502546.

48. J.-D. Boissonnat, L. J. Guibas, and S. Y. Oudot. 'Manifold reconstruction in arbitrary dimensions using witness complexes'. *Discrete & Computational Geometry* 42:1, 2009, pp. 37–70. DOI: 10.1007/s00454-009-9175-1.

49. I. Borg and P. J. F. Groenen. *Modern multidimensional scaling. Theory and applications*. 2nd ed. Springer, New York, NY, USA, 2005. DOI: 10.1007/0-387-28981-X.

50. D. Borland and R. M. Taylor II. 'Rainbow color map (still) considered harmful'. *IEEE Computer Graphics and Applications* 27:2, 2007, pp. 14–17. DOI: 10.1109/MCG.2007.46.

51. K. Borsuk. 'On the imbedding of systems of compacta in simplicial complexes'. *Fundamenta Mathematicæ* 35:1, 1948, pp. 217–234.

52. G. E. Box and D. R. Cox. 'An analysis of transformations'. *Journal of the Royal Statistical Society. Series B: Statistical Methodology* 26:2, 1964, pp. 211–252.

53. G. E. Bredon. *Topology and geometry*. Graduate Texts in Mathematics 139. Springer, New York, NY, USA, 2002. DOI: 10.1007/978-1-4757-6848-0.

54. L. Breiman. 'Heuristics of instability and stabilization in model selection'. *The Annals of Statistics* 24:6, 1996, pp. 2350–2383. DOI: 10.1214/aos/1032181158.

55. P.-T. Bremer, D. Maljovec, A. Saha, B. Wang, J. Gaffney, B. K. Spears, and V. Pascucci. 'ND^{2}AV: N-dimensional data analysis and visualization analysis for the National Ignition Campaign'. *Computing and Visualization in Science* 17:1, 2015, pp. 1–18. DOI: 10.1007/s00791-015-0241-3.

56. A. M. Bronstein, M. M. Bronstein, R. Kimmel, M. Mahmoudi, and G. Sapiro. 'A Gromov–Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching'. *International Journal of Computer Vision* 89:2, 2010, pp. 266–286. DOI: 10.1007/s11263-009-0301-6.

57. M. M. Bronstein and I. Kokkinos. 'Scale-invariant heat kernel signatures for non-rigid shape recognition'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR). Curran Associates, Inc., Red Hook, NY, USA, 2010, pp. 1704–1711. DOI: 10.1109/CVPR.2010.5539838.

58. A. E. Brouwer and W. H. Haemers. *Spectra of graphs*. Springer, New York, NY, USA, 2012. DOI: 10.1007/978-1-4614-1939-6.

59. S. Bruckner and T. Möller. 'Result-driven exploration of simulation parameter spaces for visual effects design'. *IEEE Transactions on Visualization and Computer Graphics* 16:6, 2010, pp. 1467–1475. DOI: 10.1109/TVCG.2010.190.

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

61. M. Buchet, F. Chazal, S. Y. Oudot, and D. R. Sheehy. 'Efficient and robust persistent homology for measures'. *Computational Geometry* 58, 2016, pp. 70–96. DOI: 10.1016/j.comgeo.2016.07.001.

62. M. D. Buhmann. 'Radial basis functions'. *Acta Numerica* 9, 2000, pp. 1–38.

63. A. Buja, J. A. McDonald, J. Michalak, and W. Stuetzle. 'Interactive data visualization using focusing and linking'. In: *Proceedings of the 2nd Annual Conference on Visualization*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1991, pp. 156–163. DOI: 10.1109/VISUAL.1991.175794.

64. C. S. Bullock III and D. W. Brady. 'Party, constituency, and roll-call voting in the U. S. Senate'. *Legislative Studies Quarterly* 8:1, 1983, pp. 29–43.

65. S. S. Cairns. 'Triangulation of the manifold of class one'. *Bulletin of the American Mathematical Society* 41:8, 1935, pp. 549–552.

66. J. Cardinal, S. Collette, and S. Langerman. 'Empty region graphs'. *Computational Geometry* 42:3, 2009, pp. 183–195. DOI: 10.1016/j.comgeo.2008.09.003.

67. G. Carlsson. 'Topological pattern recognition for point cloud data'. *Acta Numerica* 23, 2014, pp. 289–368. DOI: 10.1017/S0962492914000051.

68. G. Carlsson and F. Mémoli. 'Characterization, stability and convergence of hierarchical clustering methods'. *Journal of Machine Learning Research* 11, 2010, pp. 1425–1470.

69. G. Carlsson and F. Mémoli. 'Classifying clustering schemes'. *Foundations of Computational Mathematics* 13:2, 2013, pp. 221–252. DOI: 10.1007/s10208-012-9141-9.

70. G. Carlsson, G. Singh, and A. J. Zomorodian. 'Computing multidimensional persistence'. *Journal of Computational Geometry* 1:1, 2010, pp. 72–100.

71. G. Carlsson and A. J. Zomorodian. 'The theory of multidimensional persistence'. *Discrete & Computational Geometry* 42:1, 2009, pp. 71–93. DOI: 10.1007/s00454-009-9176-0.

72. G. Carlsson, A. J. Zomorodian, A. Collins, and L. J. Guibas. 'Persistence barcodes for shapes'. *International Journal of Shape Modeling* 11:2, 2005, pp. 149–187. DOI: 10.1142/S0218654305000761.

73. D. B. Carr. 'Looking at large data sets using binned data plots'. In: *Computing and Graphics in Statistics*. Ed. by A. Buja and P. A. Tukey. Springer, New York, NY, USA, 1991, pp. 7–39.

74. D. B. Carr, R. J. Littlefield, W. L. Nicholson, and J. S. Littlefield. 'Scatterplot matrix techniques for large N'. *Journal of the American Statistical Association* 82:398, 1987, pp. 424–436. DOI: 10.2307/2289444.

75. D. B. Carr, A. R. Olsen, and D. White. 'Hexagon mosaic maps for display of univariate and bivariate geographical data'. *Cartography and Geographic Information Systems* 19:4, 1992, pp. 228–236. DOI: 10.1559/152304092783721231.

76. H. Carr and D. Duke. 'Joint contour nets'. *IEEE Transactions on Visualization and Computer Graphics* 20:8, 2014, pp. 1100–1113. DOI: 10.1109/TVCG.2013.269.

77. H. Carr and D. Duke. 'Joint contour nets: Computation and properties'. In: *IEEE Pacific Visualization Symposium* (PacificVis). Curran Associates, Inc., Red Hook, NY, USA, 2013, pp. 161–168. DOI: 10.1109/PacificVis.2013.6596141.

78. H. Carr, Z. Geng, J. Tierny, A. Chattopadhyay, and A. Knoll. 'Fiber surfaces: Generalizing isosurfaces to bivariate data'. *Computer Graphics Forum* 34:3, 2015, pp. 241–250. DOI: 10.1111/cgf.12636.

79. H. Carr, J. Snoeyink, and U. Axen. 'Computing contour trees in all dimensions'. *Computational Geometry* 24:2, 2003, pp. 75–94. DOI: 10.1016/S0925-7721(02)00093-7.

80. H. Carr, J. Snoeyink, and M. van de Panne. 'Flexible isosurfaces: Simplifying and displaying scalar topology using the contour tree'. *Computational Geometry* 43:1, 2010, pp. 42–58. DOI: 10.1016/j.comgeo.2006.05.009.

81. M. Carrière and S. Oudot. 'Structure and stability of the 1-dimensional Mapper'. In: *32nd International Symposium on Computational Geometry*. Ed. by S. Fekete and A. Lubiw. Vol. 51. Leibniz International Proceedings in Informatics. Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2016, 25:1–25:16. DOI: 10.4230/LIPIcs.SoCG.2016.25.

82. N. J. Cavanna, M. Jahanseir, and D. R. Sheehy. 'A geometric perspective on sparse filtrations'. In: *Proceedings of the 27th Canadian Conference on Computational Geometry*. 2015, pp. 116–121.

83. A. Cerri, B. Di Fabio, G. Jabłoński, and F. Medri. 'Comparing shapes through multiscale approximations of the matching distance'. *Computer Vision and Image Understanding* 121, 2014, pp. 43–56. DOI: 10.1016/j.cviu.2013.11.004.

84. A. Cerri and P. Frosini. 'Advances in multidimensional size theory'. *Image Analysis & Stereology* 29:1, 2011, pp. 19–26. DOI: 10.5566/ias.v29.p19-26.

85. A. Cerri and C. Landi. 'The persistence space in multidimensional persistent homology'. In: *Discrete Geometry for Computer Imagery*. Ed. by R. González-Díaz, M.-J. Jiménez, and B. Medrano. Lecture Notes in Computer Science 7749. Springer, Cham, Switzerland, 2013, pp. 180–191. DOI: 10.1007/978-3-642-37067-0_16.

86. J. M. Chambers, W. S. Cleveland, B. Kleiner, and P. A. Tukey. *Graphical methods for data analysis*. Wadsworth & Brooks/Cole Publishing Company, Pacific Grove, CA, USA, 1983.

87. J. M. Chan, G. Carlsson, and R. Rabadan. 'Topology of viral evolution'. *Proceedings of the National Academy of Sciences of the United States of America* (PNAS) 110:46, 2013, pp. 18566–18571. DOI: 10.1073/pnas.1313480110.

88. F. Chazal, D. Cohen-Steiner, L. J. Guibas, F. Mémoli, and S. Y. Oudot. 'Gromov–Hausdorff stable signatures for shapes using persistence'. *Computer Graphics Forum* 28:5, 2009, pp. 1393–1403. DOI: 10.1111/j.1467-8659.2009.01516.x.

89. F. Chazal, D. Cohen-Steiner, and Q. Mérigot. 'Geometric inference for probability measures'. *Foundations of Computational Mathematics* 11:6, 2011, pp. 733–751. DOI: 10.1007/s10208-011-9098-0.

90. F. Chazal, B. T. Fasy, F. Lecci, A. Rinaldo, A. Singh, and L. Wasserman. 'On the bootstrap for persistence diagrams and landscapes'. *Modeling and Analysis of Information Systems (Моделирование и Анализ Информационных Систем)* 20:6, 2013, pp. 111–120.

91. F. Chazal, L. J. Guibas, S. Y. Oudot, and P. Skraba. 'Persistence-based clustering in Riemannian manifolds'. *Journal of the ACM* 60:6, 2013, 41:1–41:38. DOI: 10.1145/2535927.

92. F. Chazal, V. de Silva, and S. Y. Oudot. 'Persistence stability for geometric complexes'. *Geometriæ Dedicata* 173:1, 2014, pp. 193–214. DOI: 10.1007/s10711-013-9937-z.

93. C. Chen and H. Edelsbrunner. 'Diffusion runs low on persistence fast'. In: *Proceedings of the IEEE International Conference on Computer Vision* (ICCV). Curran Associates, Inc., Red Hook, NY, USA, 2011, pp. 423–430. DOI: 10.1109/ICCV.2011.6126271.

94. C. Chen and D. Freedman. 'Hardness results for homology localization'. *Discrete & Computational Geometry* 45:3, 2011, pp. 425–448. DOI: 10.1007/s00454-010-9322-8.

95. C. Chen and D. Freedman. 'Measuring and computing natural generators for homology groups'. *Computational Geometry* 43:2, 2010, pp. 169–181. DOI: 10.1016/j.comgeo.2009.06.004.

96. L. Chen and A. Buja. 'Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis'. *Journal of the American Statistical Association* 104:485, 2009, pp. 209–219. DOI: 10.1198/jasa.2009.0111.

97. M. Chen and H. Jänicke. 'An information-theoretic framework for visualization'. *IEEE Transactions on Visualization and Computer Graphics* 16:6, 2010, pp. 1206–1215. DOI: 10.1109/TVCG.2010.132.

98. Y.-C. Chen, D. Wang, A. Rinalo, and L. Wasserman. 'Statistical analysis of persistence intensity functions'. Preprint. 2015. URL: http://arxiv.org/abs/1510.02502.

99. Y. Cheng. 'Mean shift, mode seeking, and clustering'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17:8, 1995, pp. 790–799. DOI: 10.1109/34.400568.

100. D. Child. *The essentials of factor analysis*. 3rd ed. Continuum International Publishing Group, London, England, 2006.

101. H. Chintakunta, T. Gentimis, R. González-Díaz, M.-J. Jiménez, and H. Krim. 'An entropy-based persistence barcode'. *Pattern Recognition* 48:2, 2015, pp. 391–401. DOI: 10.1016/j.patcog.2014.06.023.

102. F. R. K. Chung. *Spectral graph theory*. Regional Conference Series in Mathematics 92. American Mathematical Society, Providence, RI, USA, 1997.

103. D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. 'Extending persistence using Poincaré and Lefschetz duality'. *Foundations of Computational Mathematics* 9:1, 2009, pp. 79–103. DOI: 10.1007/s10208-008-9027-z.

104. D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. 'Stability of persistence diagrams'. *Discrete & Computational Geometry* 37:1, 2007, pp. 103–120. DOI: 10.1007/s00454-006-1276-5.

105. D. Cohen-Steiner, H. Edelsbrunner, J. Harer, and Y. Mileyko. 'Lipschitz functions have L_p-stable persistence'. *Foundations of Computational Mathematics* 10:2, 2010, pp. 127–139. DOI: 10.1007/s10208-010-9060-6.

106. D. Cohen-Steiner, H. Edelsbrunner, and D. Morozov. 'Vines and vineyards by updating persistence in linear time'. In: *Proceedings of the 22nd Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 2006, pp. 119–126. DOI: 10.1145/1137856.1137877.

107. R. R. Coifman and S. Lafon. 'Diffusion maps'. *Applied and Computational Harmonic Analysis* 21:1, 2006, pp. 5–30. DOI: 10.1016/j.acha.2006.04.006.

108. K. Cole-McLaughlin, H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. 'Loops in Reeb graphs of 2-manifolds'. In: *Proceedings of the 19th Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 2003, pp. 344–350. DOI: 10.1145/777792.777844.

109. A. Collins, A. J. Zomorodian, G. Carlsson, and L. J. Guibas. 'A barcode shape descriptor for curve point cloud data'. *Computers & Graphics* 28:6, 2004, pp. 881–894. DOI: 10.1016/j.cag.2004.08.015.

110. M. Collins, S.-I. An, W. Cai, A. Ganachaud, E. Guilyardi, F.-F. Jin, M. Jochum, M. Lengaigne, S. Power, A. Timmermann, G. Vecchi, and A. Wittenberg. 'The impact of global warming on the tropical Pacific Ocean and El Niño'. *Nature Geoscience* 3:6, 2010, pp. 391–397. DOI: 10.1038/ngeo868.

111. D. Cook, A. Buja, E. Lee, and H. Wickham. 'Grand tours, projection pursuit guided tours and manual controls'. In: *Handbook of Data Visualization*. Ed. by C. Chen, W. Härdle, and A. Unwin. Springer, Heidelberg, Germany, 2008, pp. 295–314. DOI: 10.1007/978-3-540-33037-0_13.

112. D. Cook and D. F. Swayne. *Interactive and dynamic graphics for data analysis. With R and GGobi*. Springer, New York, NY, USA, 2007. DOI: 10.1007/978-0-387-71762-3.

113. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. 3rd ed. MIT Press, Cambridge, MA, USA, 2009.

114. C. D. Correa and P. Lindstrom. 'Locally-scaled spectral clustering using empty region graphs'. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD). ACM Press, New York, NY, USA, 2012, pp. 1330–1338. DOI: 10.1145/2339530.2339736.

115. C. D. Correa and P. Lindstrom. 'Towards robust topology of sparsely sampled data'. *IEEE Transactions on Visualization and Computer Graphics* 17:12, 2011, pp. 1852–1861. DOI: 10.1109/TVCG.2011.245.

116. C. D. Correa, P. Lindstrom, and P.-T. Bremer. 'Topological spines: A structure-preserving visual representation of scalar fields'. *IEEE Transactions on Visualization and Computer Graphics* 17:12, 2011, pp. 1842–1851. DOI: 10.1109/TVCG.2011.244.

117. D. Cox and V. Isham. *Point processes*. Monographs on Statistics and Applied Probability 12. Chapman & Hall/CRC, Boca Raton, FL, USA, 1980.

118. T. F. Cox and T. Lewis. 'A conditioned distance ratio method for analyzing spatial patterns'. *Biometrika* 63:3, 1976, pp. 483–491. DOI: 10.2307/2335725.

119. N. A. C. Cressie. *Statistics for spatial data*. John Wiley & Sons, Ltd., New York, NY, USA, 1993. DOI: 10.1002/9781119115151.

120. G. Damiand, R. González-Díaz, and S. Peltier. 'Removal operations in nD generalized maps for efficient homology computation'. In: *Computational Topology in Image Context*. Ed. by M. Ferri, P. Frosini, C. Landi, A. Cerri, and B. Di Fabio. Springer, Heidelberg, Germany, 2012, pp. 20–29. DOI: 10.1007/978-3-642-30238-1_3.

121. S. Dantchev and I. P. Ivrissimtzis. 'Efficient construction of the Čech complex'. *Computers & Graphics* 36:6, 2012, pp. 708–713. DOI: 10.1016/j.cag.2012.02.016.

122. J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. 'Information-theoretic metric learning'. In: *Proceedings of the 24th International Conference on Machine Learning*. ACM Press, New York, NY, USA, 2007, pp. 209–216. DOI: 10.1145/1273496.1273523.

123. L. De Floriani and A. Hui. 'Data structures for simplicial complexes: An analysis and a comparison'. In: *Eurographics Symposium on Geometry Processing*. Ed. by M. Desbrun and H. Pottmann. The Eurographics Association, 2005. DOI: 10.2312/SGP/SGP05/119-128.

124. L. De Floriani, A. Hui, D. Panozzo, and D. Canino. 'A dimension-independent data structure for simplicial complexes'. In: *Proceedings of the 19th International Meshing Roundtable*. Ed. by S. Shontz. Springer, Heidelberg, Germany, 2010, pp. 403–420. DOI: 10.1007/978-3-642-15414-0_24.

125. R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart. 'The Mahalanobis distance'. *Chemometrics and Intelligent Laboratory Systems* 50:1, 2000, pp. 1–18. DOI: 10.1016/S0169-7439(99)00047-7.

126. A. Deckard, R. C. Anafi, J. B. Hogenesch, S. B. Haase, and J. Harer. 'Design and analysis of large-scale biological rhythm studies: A comparison of algorithms for detecting periodic signals in biological data'. *Bioinformatics* 29:24, 2013, pp. 3174–3180. DOI: 10.1093/bioinformatics/btt541.

127. T. K. Dey, F. Fan, and Y. Wang. 'An efficient computation of handle and tunnel loops via Reeb graphs'. *ACM Transactions on Graphics* 32:4, 2013, 32:1–32:10. DOI: 10.1145/2461912.2462017.

128. P. Diaconis, S. Holmes, and M. Shahshahani. 'Sampling from a manifold'. In: *Advances in Modern Statistical Theory and Applications. A Festschrift in honor of Morris L. Eaton*. Collections 10. Institute of Mathematical Statistics, Beachwood, OH, USA, 2013, pp. 102–125. DOI: 10.1214/12-IMSCOLL1006.

129. P. Diaconis and M. Shahshahani. 'The subgroup algorithm for generating uniform random variables'. *Probability in the Engineering and Informational Sciences* 1:01, 1987, pp. 15–32. DOI: 10.1017/S0269964800000255.

130. H. Doleisch and H. Hauser. 'Smooth brushing for focus+context visualization of simulation data in 3D'. *Journal of WSCG* 10:1–3, 2002, pp. 147–154.

131. D. L. Donoho. *50 years of data science*. Talk at the Tukey Centennial Workshop. Princeton University, 2015.

132. D. L. Donoho and C. Grimes. 'Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data'. *Proceedings of the National Academy of Sciences of the United States of America* (PNAS) 100:10, 2003, pp. 5591–5596. DOI: 10.1073/pnas.1031596100.

133. D. L. Donoho and C. Grimes. 'Image manifolds which are isometric to Euclidean space'. *Journal of Mathematical Imaging and Vision* 23:1, 2005, pp. 5–24. DOI: 10.1007/s10851-005-4965-4.

134. H. Doraiswamy and V. Natarajan. 'Efficient algorithms for computing Reeb graphs'. *Computational Geometry* 42:6–7, 2009, pp. 606–616. DOI: 10.1016/j.comgeo.2008.12.003.

135. T. A. Driscoll and L. N. Trefethen. *Schwarz–Christoffel mapping*. Cambridge Monographs on Applied and Computational Mathematics 8. Cambridge University Press, Cambridge, England, 2002.

136. H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik. 'Support vector regression machines'. In: *Advances in Neural Information Processing Systems 9* (NIPS). Ed. by M. C. Mozer, M. I. Jordan, and T. Petsche. MIT Press, Cambridge, MA, USA, 1997, pp. 155–161.

137. D. Duke, H. Carr, A. Knoll, N. Schunck, H. A. Nam, and A. Staszczak. 'Visualizing nuclear scission through a multifield extension of topological analysis'. *IEEE Transactions on Visualization and Computer Graphics* 18:12, 2012, pp. 2033–2040. DOI: 10.1109/TVCG.2012.287.

138. J.-G. Dumas, F. Heckenbach, D. Saunders, and V. Welker. 'Computing simplicial homology based on efficient Smith normal form algorithms'. In: *Algebra, Geometry, and Software Systems*. Ed. by M. Joswig and N. Takayama. Springer, Heidelberg, Germany, 2003, pp. 177–206. DOI: 10.1007/978-3-662-05148-1_10.

139. J. C. Dunn. 'A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters'. *Journal of Cybernetics* 3:3, 1973, pp. 32–57. DOI: 10.1080/01969727308546046.

140. C. Eckart and G. Young. 'The approximation of one matrix by another of lower rank'. *Psychometrika* 1:3, 1936, pp. 211–218. DOI: 10.1007/BF02288367.

141. H. Edelsbrunner and J. Harer. *Computational topology: An introduction*. American Mathematical Society, Providence, RI, USA, 2010.

142. H. Edelsbrunner and J. Harer. 'Persistent homology—a survey'. In: *Surveys on discrete and computational geometry: Twenty years later*. Ed. by J. E. Goodman, J. Pach, and R. Pollack. Contemporary Mathematics 453. American Mathematical Society, Providence, RI, USA, 2008, pp. 257–282.

143. H. Edelsbrunner, J. Harer, A. Mascarenhas, V. Pascucci, and J. Snoeyink. 'Time-varying Reeb graphs for continuous space-time data'. *Computational Geometry* 41:3, 2008, pp. 149–166. DOI: 10.1016/j.comgeo.2007.11.001.

144. H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. 'Morse–Smale complexes for piecewise linear 3-manifolds'. In: *Proceedings of the 19th Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 2003, pp. 361–370. DOI: 10.1145/777792.777846.

145. H. Edelsbrunner, J. Harer, and A. K. Patel. 'Reeb spaces of piecewise linear mappings'. In: *Proceedings of the 24th Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 2008, pp. 242–250. DOI: 10.1145/1377676.1377720.

146. H. Edelsbrunner, J. Harer, and A. J. Zomorodian. 'Hierarchical Morse–Smale complexes for piecewise linear 2-manifolds'. *Discrete & Computational Geometry* 30:1, 2003, pp. 87–107. DOI: 10.1007/s00454-003-2926-5.

147. H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel. 'On the shape of a set of points in the plane'. *IEEE Transactions on Information Theory* 29:4, 1983, pp. 551–559. DOI: 10.1109/TIT.1983.1056714.

148. H. Edelsbrunner, D. Letscher, and A. J. Zomorodian. 'Topological persistence and simplification'. *Discrete & Computational Geometry* 28:4, 2002, pp. 511–533. DOI: 10.1007/s00454-002-2885-2.

149. H. Edelsbrunner and D. Morozov. 'Persistent homology: Theory and practice'. In: *European Congress of Mathematics*. Ed. by R. Latała, A. Ruciński, P. Strzelecki, J. Świątkowski, D. Wrzosek, and P. Zakrzewski. European Mathematical Society Publishing House, Zürich, Switzerland, 2014. DOI: 10.4171/120-1/3.

150. H. Edelsbrunner, D. Morozov, and V. Pascucci. 'Persistence-sensitive simplification of functions on 2-manifolds'. In: *Proceedings of the 22nd Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 2006, pp. 127–134. DOI: 10.1145/1137856.1137878.

151. H. Edelsbrunner and E. P. Mücke. 'Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms'. *ACM Transactions on Graphics* 9:1, 1990, pp. 66–104. DOI: 10.1145/77635.77639.

152. H. Edelsbrunner and E. P. Mücke. 'Three-dimensional alpha shapes'. *ACM Transactions on Graphics* 13:1, 1994, pp. 43–72. DOI: 10.1145/174462.156635.

153. A. Efrat, A. Itai, and M. J. Katz. 'Geometry helps in bottleneck matching and related problems'. *Algorithmica* 31:1, 2001, pp. 1–28. DOI: 10.1007/s00453-001-0016-8.

154. G. Ellis and A. Dix. 'Enabling automatic clutter reduction in parallel coordinate plots'. *IEEE Transactions on Visualization and Computer Graphics* 12:5, 2006, pp. 717–724. DOI: 10.1109/TVCG.2006.138.

155. N. Elmqvist, P. Dragicevic, and J.-D. Fekete. 'Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation'. *IEEE Transactions on Visualization and Computer Graphics* 14:6, 2008, pp. 1539–1148. DOI: 10.1109/TVCG.2008.153.

156. N. Elmqvist and J.-D. Fekete. 'Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines'. *IEEE Transactions on Visualization and Computer Graphics* 16:3, 2010, pp. 439–454. DOI: 10.1109/TVCG.2009.84.

157. N. Elmqvist, J. Stasko, and P. Tsigas. 'DataMeadow: A visual canvas for analysis of large-scale multivariate data'. *Information Visualization* 7:1, 2008, pp. 18–33. DOI: 10.1057/palgrave.ivs.9500170.

158. D. Eppstein. 'Spanning trees and spanners'. In: *Handbook of Computational Geometry*. Ed. by J. Sack and J. Urrutia. North-Holland Publishing Company, Amsterdam, Netherlands, 2000, pp. 425–461. DOI: 10.1016/B978-044482537-7/50010-3.

159. J. Erickson and K. Whittlesey. 'Greedy optimal homotopy and homology generators'. In: *Proceedings of the 16th Annual ACM–SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2005, pp. 1038–1046.

160. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. 'A density-based algorithm for discovering clusters in large spatial databases with noise'. In: *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining* (KDD). Ed. by E. Simoudis, J. Han, and U. Fayyad. AAAI Press, Palo Alto, CA, USA, 1996, pp. 226–231.

161. J. Fangerau, B. Höckendorf, B. Rieck, C. Heine, J. Wittbrodt, and H. Leitte. 'Interactive similarity analysis and error detection in large tree collections'. In: *Visualization in Medicine and Life Sciences III. Towards making an impact*. Ed. by L. Linsen, B. Hamann, and H.-C. Hege. Mathematics and Visualization. Springer, Cham, Switzerland, 2016, pp. 287–307. DOI: 10.1007/978-3-319-24523-2_13.

162. M. Farach-Colton and P. Indyk. 'Approximate nearest neighbor algorithms for Hausdorff metrics via embeddings'. In: *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*. Curran Associates, Inc., Red Hook, NY, USA, 1999, pp. 171–179. DOI: 10.1109/SFFCS.1999.814589.

163. B. T. Fasy, F. Lecci, A. Rinaldo, L. Wasserman, S. Balakrishnan, and A. Singh. 'Confidence sets for persistence diagrams'. *The Annals of Statistics* 42:6, 2014, pp. 2301–2339. DOI: 10.1214/14-AOS1252.

164. K.-C. Feng, C. Wang, H.-W. Shen, and T.-Y. Lee. 'Coherent time-varying graph drawing with multifocus+context interaction'. *IEEE Transactions on Visualization and Computer Graphics* 18:8, 2012, pp. 1330–1342. DOI: 10.1109/TVCG.2011.128.

165. K. Fischer, B. Gärtner, and M. Kutz. 'Fast smallest-enclosing-ball computation in high dimensions'. In: *Algorithms — ESA 2003*. Ed. by G. D. Battista and U. Zwick. Lecture Notes in Computer Science 2832. Springer, Heidelberg, Germany, 2003, pp. 630–641. DOI: 10.1007/978-3-540-39658-1_57.

166. S. Fortune. 'Voronoi diagrams and Delaunay triangulations'. In: *Handbook of Discrete and Computational Geometry*. Ed. by J. E. Goodman and J. O'Rourke. 2nd ed. Chapman & Hall/CRC, Boca Raton, FL, USA, 2004. Chap. 23, pp. 513–528.

167. E. B. Fowlkes and C. L. Mallows. 'A method for comparing two hierarchical clusterings'. *Journal of the American Statistical Association* 78:383, 1983, pp. 553–569. DOI: 10.1080/01621459.1983.10478008.

168. D. Freedman and P. Diaconis. 'On the histogram as a density estimator: L_2 theory'. *Probability Theory and Related Fields* 57:4, 1981, pp. 453–476. DOI: 10.1007/BF01025868.

169. L. C. Freeman. 'A set of measures of centrality based on betweenness'. *Sociometry* 40:1, 1977, pp. 35–41. DOI: 10.2307/3033543.

170. L. C. Freeman, S. P. Borgatti, and D. R. White. 'Centrality in valued graphs: A measure of betweenness based on network flow'. *Social Networks* 13:2, 1991, pp. 141–154. DOI: 10.1016/0378-8733(91)90017-N.

171. J. H. Friedman and J. W. Tukey. 'A projection pursuit algorithm for exploratory data analysis'. *IEEE Transactions on Computers* C-23:9, 1974, pp. 881–890. DOI: 10.1109/T-C.1974.224051.

172. P. Frosini. 'A distance for similarity classes of submanifolds of a Euclidean space'. *Bulletin of the Australian Mathematical Society* 42:3, 1990, pp. 407–415. DOI: 10.1017/S0004972700028574.

173. K. R. Gabriel and R. R. Sokal. 'A new statistical approach to geographic variation analysis'. *Systematic Biology* 18:3, 1969, pp. 259–278. DOI: 10.2307/2412323.

174. E. R. Gansner, Y. Koren, and S. North. 'Graph drawing by stress majorization'. In: *Graph drawing*. Ed. by J. Pach. Lecture Notes in Computer Science 3383. Springer, Heidelberg, Germany, 2005, pp. 239–250. DOI: 10.1007/978-3-540-31843-9_25.

175. B. Gärtner. 'Fast and robust smallest enclosing balls'. In: *Algorithms — ESA '99*. Ed. by J. Nešetřil. Lecture Notes in Computer Science 1643. Springer, Heidelberg, Germany, 1999, pp. 325–338. DOI: 10.1007/3-540-48481-7_29.

176. S. Gerber, P.-T. Bremer, V. Pascucci, and R. Whitaker. 'Visual exploration of high dimensional scalar functions'. *IEEE Transactions on Visualization and Computer Graphics* 16:6, 2010, pp. 1271–1280. DOI: 10.1109/TVCG.2010.213.

177. R. Ghrist. 'Barcodes: The persistent topology of data'. *Bulletin of the American Mathematical Society* 45:1, 2008, pp. 61–75. DOI: 10.1090/S0273-0979-07-01191-3.

178. R. Ghrist. 'Three examples of applied and computational homology'. *Nieuw Archief voor Wiskunde* 9:2, 2008, pp. 122–125.

179. J. Giesen and M. John. 'The flow complex: A data structure for geometric modeling'. *Computational Geometry* 39:3, 2008, pp. 178–190. DOI: 10.1016/j.comgeo.2007.01.002.

180. L. Gosink, K. Bensema, T. Pulsipher, H. Obermaier, M. Henry, H. Childs, and K. Joy. 'Characterizing and visualizing predictive uncertainty in numerical ensembles through Bayesian model averaging'. *IEEE Transactions on Visualization and Computer Graphics* 19:12, 2013, pp. 2703–2712. DOI: 10.1109/TVCG.2013.138.

181. A. Grigor'yan. 'Heat kernels on weighted manifolds and applications'. In: *The Ubiquitous Heat Kernel*. Ed. by J. Jorgenson and L. Walling. Contemporary Mathematics 398. American Mathematical Society, Providence, RI, USA, 2006, pp. 93–191. DOI: 10.1090/conm/398.

182. E. Grinspun, Y. Gingold, J. Reisman, and D. Zorin. 'Computing discrete shape operators on general meshes'. *Computer Graphics Forum* 25:3, 2006, pp. 547–556. DOI: 10.1111/j.1467-8659.2006.00974.x.

183. L. J. Guibas and S. Y. Oudot. 'Reconstruction using witness complexes'. *Discrete & Computational Geometry* 40:3, 2008, pp. 325–356. DOI: 10.1007/s00454-008-9094-6.

184. A. Gyulassy, P.-T. Bremer, B. Hamann, and V. Pascucci. 'A practical approach to Morse–Smale complex computation: Scalability and generality'. *IEEE Transactions on Visualization and Computer Graphics* 14:6, 2008, pp. 1619–1626. DOI: 10.1109/TVCG.2008.110.

185. A. Gyulassy, A. Knoll, K. C. Lau, B. Wang, P.-T. Bremer, M. E. Papka, L. A. Curtiss, and V. Pascucci. 'Interstitial and interlayer ion diffusion geometry extraction in graphitic nanosphere battery materials'. *IEEE Transactions on Visualization and Computer Graphics* 22:1, 2016, pp. 916–925. DOI: 10.1109/TVCG.2015.2467432.

186. A. Gyulassy, V. Natarajan, V. Pascucci, P.-T. Bremer, and B. Hamann. 'A topological approach to simplification of three-dimensional scalar functions'. *IEEE Transactions on Visualization and Computer Graphics* 12:4, 2006, pp. 474–484. DOI: 10.1109/TVCG.2006.57.

187. M. Halkidi, Y. Batistakis, and M. Vazirgiannis. 'On clustering validation techniques'. *Journal of Intelligent Information Systems* 17:2–3, 2001, pp. 107–145. DOI: 10.1023/A:1012801612483.

188. S. Har-Peled and M. Mendel. 'Fast construction of nets in low-dimensional metrics and their applications'. *SIAM Journal on Computing* 35:5, 2006, pp. 1148–1184. DOI: 10.1137/S0097539704446281.

189. J. A. Hartigan. 'Consistency of single linkage for high-density clusters'. *Journal of the American Statistical Association* 76:374, 1981, pp. 388–394. DOI: 10.1080/01621459.1981.10477658.

190. J. A. Hartigan and B. Kleiner. 'Mosaics for contingency tables'. In: *Computer Science and Statistics: Proceedings of the 13th Symposium on the Interface*. Ed. by W. F. Eddy. Springer, New York, NY, USA, 1981, pp. 268–273. DOI: 10.1007/978-1-4613-9464-8_37.

191. W. Harvey and Y. Wang. 'Topological landscape ensembles for visualization of scalar-valued functions'. *Computer Graphics Forum* 29:3, 2010, pp. 993–1002. DOI: 10.1111/j.1467-8659.2009.01706.x.

192. A. Hatcher. *Algebraic topology*. Cambridge University Press, Cambridge, England, 2002.

193. S. Havre, E. Hetzler, P. Whitney, and L. Nowell. 'ThemeRiver: Visualizing thematic changes in large document collections'. *IEEE Transactions on Visualization and Computer Graphics* 8:1, 2002, pp. 9–20. DOI: 10.1109/2945.981848.

194. S. P. Hayes, L. J. Mangum, J. Picaut, A. Sumi, and K. Takeuchi. 'TOGA–TAO: A moored array for real-time measurements in the Tropical Pacific Ocean'. *Bulletin of the American Meteorological Society* 72:3, 1991, pp. 339–347. DOI: 10.1175/1520-0477(1991)072<0339:TTAMAF>2.0.CO;2.

195. C. Heine, H. Leitte, M. Hlawitschka, F. Iuricich, L. De Floriani, G. Scheuermann, H. Hagen, and C. Garth. 'A survey of topology-based methods in visualization'. *Computer Graphics Forum* 35:3, 2016, pp. 643–667. DOI: 10.1111/cgf.12933.

196. C. Heine, D. Schneider, H. Carr, and G. Scheuermann. 'Drawing contour trees in the plane'. *IEEE Transactions on Visualization and Computer Graphics* 17:11, 2011, pp. 1599–1611. DOI: 10.1109/TVCG.2010.270.

197. J. Heinrich and D. Weiskopf. 'Continuous parallel coordinates'. *IEEE Transactions on Visualization and Computer Graphics* 15:6, 2009, pp. 1531–1538. DOI: 10.1109/TVCG.2009.131.

198. A. Hinneburg, C. C. Aggarwal, and D. A. Keim. 'What is the nearest neighbor in high dimensional spaces?' In: *Proceedings of the 26th International Conference on Very Large Data Bases*. Ed. by A. E. Abbadi, M. L. Brodie, S. Chakravarthy, U. Dayal, N. Kamel, G. Schlageter, and K. Whang. Morgan Kaufmann Publishers, Burlington, MA, USA, 2000, pp. 506–515.

199. A. Hinneburg and D. A. Keim. 'An efficient approach to clustering in large multimedia databases with noise'. In: *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining* (KDD). Ed. by R. Agrawal and P. Stolorz. AAAI Press, Palo Alto, CA, USA, 1998, pp. 58–65.

200. G. E. Hinton, P. Dayan, and M. Revow. 'Modeling the manifolds of images of handwritten digits'. *IEEE Transactions on Neural Networks* 8:1, 1997, pp. 65–74. DOI: 10.1109/72.554192.

201. D. Holten and J. J. van Wijk. 'Evaluation of cluster identification performance for different PCP variants'. *Computer Graphics Forum* 29:3, 2010, pp. 793–802. DOI: 10.1111/j.1467-8659.2009.01666.x.

202. K. Hornbæk, B. B. Bederson, and C. Plaisant. 'Navigation patterns and usability of zoomable user interfaces with and without an overview'. *ACM Transactions on Computer–Human Interaction* (TOCHI) 9:4, 2002, pp. 362–389. DOI: 10.1145/586081.586086.

203. H. Hotelling. 'The generalization of Student's ratio'. *The Annals of Mathematical Statistics* 2:3, 1931, pp. 360–378. DOI: 10.1214/aoms/1177732979.

204. A. S. Householder. 'Unitary triangularization of a nonsymmetric matrix'. *Journal of the ACM* 5:4, 1958, pp. 339–342. DOI: 10.1145/320941.320947.

205. L. Hubert and P. Arabie. 'Comparing partitions'. *Journal of Classification* 2:1, 1985, pp. 193–218. DOI: 10.1007/BF01908075.

206. L. Hubert and J. Schultz. 'Quadratic assignment as a general data analysis strategy'. *British Journal of Mathematical and Statistical Psychology* 29:2, 1976, pp. 190–241. DOI: 10.1111/j.2044-8317.1976.tb00714.x.

207. L. Hüttenberger, C. Heine, H. Carr, G. Scheuermann, and C. Garth. 'Towards multifield scalar topology based on Pareto optimality'. *Computer Graphics Forum* 32:3–3, 2013, pp. 341–350. DOI: 10.1111/cgf.12121.

208. P. Indyk. 'Nearest neighbors in high-dimensional spaces'. In: *Handbook of Discrete and Computational Geometry*. Ed. by J. E. Goodman and J. O'Rourke. 2nd ed. Chapman & Hall/CRC, Boca Raton, FL, USA, 2004. Chap. 39, pp. 877–892.

209. S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner, and T. Möller. 'DimStiller: Workflows for dimensional analysis and reduction'. In: *IEEE Symposium on Visual Analytics Science and Technology* (VAST). Ed. by A. MacEachren and S. Miksch. Curran Associates, Inc., Red Hook, NY, USA, 2010, pp. 3–10. DOI: 10.1109/VAST.2010.5652392.

210. A. Inselberg. 'The plane with parallel coordinates'. *The Visual Computer* 1:2, 1985, pp. 69–91. DOI: 10.1007/BF01898350.

211. A. K. Jain. 'Data clustering: 50 years beyond k-means'. *Pattern Recognition Letters* 31:8, 2010, pp. 651–666. DOI: 10.1016/j.patrec.2009.09.011.

212. A. Jakulin, W. Buntine, T. M. La Pira, and H. Brasher. 'Analyzing the U. S. Senate in 2003: Similarities, clusters, and blocs'. *Political Analysis* 17:3, 2009, pp. 291–310. DOI: 10.1093/pan/mpp006.

213. H. Jänicke, M. Böttinger, U. Mikolajewicz, and G. Scheuermann. 'Visual exploration of climate variability changes using wavelet analysis'. *IEEE Transactions on Visualization and Computer Graphics* 15:6, 2009, pp. 1375–1382. DOI: 10.1109/TVCG.2009.197.

214. H. Jänicke, M. Böttinger, and G. Scheuermann. 'Brushing of attribute clouds for the visualization of multivariate data'. *IEEE Transactions on Visualization and Computer Graphics* 14:6, 2008, pp. 1459–1466. DOI: 10.1109/TVCG.2008.116.

215. H. Jänicke, A. Wiebel, G. Scheuermann, and W. Kollmann. 'Multifield visualization using local statistical complexity'. *IEEE Transactions on Visualization and Computer Graphics* 13:6, 2007, pp. 1384–1391. DOI: 10.1109/TVCG.2007.70615.

216. R. A. Johnson and D. W. Wichern. *Applied multivariate statistical analysis*. 6th ed. Prentice Hall, Upper Saddle River, NJ, USA, 2007.

217. W. B. Johnson and J. Lindenstrauss. 'Extensions of Lipschitz mappings into a Hilbert space'. In: *Conference on Modern Analysis and Probability*. Contemporary Mathematics 26. American Mathematical Society, Providence, RI, USA, 1984, pp. 189–206. DOI: 10.1090/conm/026/737400.

218. P. Joia, F. Petronetto, and L. G. Nonato. 'Uncovering representative groups in multidimensional projections'. *Computer Graphics Forum* 34:3, 2015, pp. 281–290. DOI: 10.1111/cgf.12640.

219. I. T. Jolliffe. *Principal component analysis*. 2nd ed. Springer, New York, NY, USA, 2002. DOI: 10.1007/b98835.

220. M. Kerber, D. Morozov, and A. Nigmetov. 'Geometry helps to compare persistence diagrams'. In: *Proceedings of the 18th Workshop on Algorithm Engineering and Experiments* (ALENEX). Ed. by M. Goodrich and M. Mitzenmacher. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2016, pp. 103–112. DOI: 10.1137/1.9781611974317.9.

221. D. E. King. 'Dlib-ml: A machine learning toolkit'. *Journal of Machine Learning Research* 10, 2009, pp. 1755–1758.

222. J. N. Kloke. 'Methods and applications of topological data analysis'. PhD thesis. Stanford University, 2010.

223. T. Kohonen. 'Self-organized formation of topologically correct feature maps'. *Biological Cybernetics* 43:1, 1982, pp. 59–69. DOI: 10.1007/BF00337288.

224. J. B. Kruskal. 'Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis'. *Psychometrika* 29:1, 1964, pp. 1–27. DOI: 10.1007/BF02289565.

225. J. B. Kruskal. 'Nonmetric multidimensional scaling: A numerical method'. *Psychometrika* 29:2, 1964, pp. 115–129. DOI: 10.1007/BF02289694.

226. H. W. Kuhn. 'The Hungarian method for the assignment problem'. *Naval Research Logistics Quarterly* 2:1–2, 1955, pp. 83–97. DOI: 10.1002/nav.3800020109.

227. M. Kuhn and K. Johnson. *Applied predicitive modeling*. Springer, New York, NY, USA, 2013. DOI: 10.1007/978-1-4614-6849-3.

228. B. Kulis. 'Metric learning: A survey'. *Foundations and Trends in Machine Learning* 5:4, 2013, pp. 287–364. DOI: 10.1561/2200000019.

229. S. Kullback and R. A. Leibler. 'On information and sufficiency'. *The Annals of Mathematical Statistics* 22:1, 1951, pp. 79–86. DOI: 10.1214/aoms/1177729694.

230. G. Kumar and M. Garland. 'Visual exploration of complex time-varying graphs'. *IEEE Transactions on Visualization and Computer Graphics* 12:5, 2006, pp. 805–812. DOI: 10.1109/TVCG.2006.193.

231. J. Lamping and R. Rao. 'The hyperbolic browser: A focus+context technique for visualizing large hierarchies'. *Journal of Visual Languages & Computing* 7:1, 1996, pp. 33–55. DOI: 10.1006/jvlc.1996.0003.

232. S. M. Lane. *Categories for the working mathematician*. 2nd ed. Graduate Texts in Mathematics 5. Springer, New York, NY, USA, 1978. DOI: 10.1007/978-1-4757-4721-8.

233. J. Latschev. 'Vietoris–Rips complexes of metric spaces near a closed Riemannian manifold'. *Archiv der Mathematik* 77:6, 2001, pp. 522–528. DOI: 10.1007/PL00000526.

234. Y. LeCun, C. Cortes, and C. J. Burges. *The MNIST database of handwritten digits*. URL: http://yann.lecun.com/exdb/mnist.

235. H. Lee, H. Kang, M. K. Chung, B.-N. Kim, and D. S. Lee. 'Persistent brain network homology from the perspective of dendrogram'. *IEEE Transactions on Medical Imaging* 31:12, 2012, pp. 2267–2277. DOI: 10.1109/TMI.2012.2219590.

236. J. M. Lee. *Manifolds and differential geometry*. Graduate Studies in Mathematics 107. American Mathematical Society, Providence, RI, USA, 2009.

237. J. H. Lee, K. T. McDonnell, A. Zelenyuk, D. Imre, and K. Mueller. 'A structure-based distance metric for high-dimensional space exploration with multidimensional scaling'. *IEEE Transactions on Visualization and Computer Graphics* 20:3, 2014, pp. 351–364. DOI: 10.1109/TVCG.2013.101.

238. J. Lee. *Introduction to smooth manifolds*. 2nd ed. Graduate Texts in Mathematics 218. Springer, New York, NY, USA, 2012. DOI: 10.1007/978-1-4419-9982-5.

239. J. A. Lee and M. Verleysen. 'Quality assessment of dimensionality reduction: Rank-based criteria'. *Neurocomputing* 72:7–9, 2009, pp. 1431–1443. DOI: 10.1016/j.neucom.2008.12.017.

240. D. J. Lehmann, G. Albuquerque, M. Eisemann, M. Magnor, and H. Theisel. 'Selecting coherent and relevant plots in large scatterplot matrices'. *Computer Graphics Forum* 31:6, 2012, pp. 1895–1908. DOI: 10.1111/j.1467-8659.2012.03069.x.

241. D. J. Lehmann and H. Theisel. 'Optimal sets of projections of high-dimensional data'. *IEEE Transactions on Visualization and Computer Graphics* 22:1, 2016, pp. 609–618. DOI: 10.1109/TVCG.2015.2467132.

242. J. M. Lewis, L. van der Maaten, and V. de Sa. 'A behavioral investigation of dimensionality reduction'. In: *Proceedings of the 34th Annual Conference of the Cognitive Science Society*. Ed. by N. Miyake, D. Peebles, and R. P. Cooper. Cognitive Science Society, Austin, TX, USA, 2012, pp. 671–676.

243. R. H. Lewis and D. Morozov. 'Parallel computation of persistent homology using the blowup complex'. In: *Proceedings of the 27th ACM Symposium on Parallelism in Algorithms and Architectures*. ACM Press, New York, NY, USA, 2015, pp. 323–331. DOI: 10.1145/2755573.2755587.

244. A. Lex, M. Streit, C. Partl, K. Kashofer, and D. Schmalstieg. 'Comparative analysis of multidimensional, quantitative data'. *IEEE Transactions on Visualization and Computer Graphics* 16:6, 2010, pp. 1027–1035. DOI: 10.1109/TVCG.2010.138.

245. C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata. 'Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median'. *Journal of Experimental Social Psychology* 49:4, 2013, pp. 764–766. DOI: 10.1016/j.jesp.2013.03.013.

246. C. Li, M. Ovsjanikov, and F. Chazal. 'Persistence-based structural recognition'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR). Curran Associates, Inc., Red Hook, NY, USA, 2014, pp. 2003–2010. DOI: 10.1109/CVPR.2014.257.

247. M. Lichman. *UCI Machine Learning Repository*. University of California, Irvine. 2013. URL: http://archive.ics.uci.edu/ml.

248. D. Lipsky, P. Skraba, and M. Vejdemo-Johansson. 'A spectral sequence for parallelized persistence'. Preprint. 2011. URL: http://arxiv.org/abs/1112.1245.

249. S. Lisitsyn, C. Widmer, and F. J. I. Garcia. 'Tapkee: An efficient dimension reduction library'. *Journal of Machine Learning Research* 14, 2013, pp. 2355–2359.

250. G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. 'Robust recovery of subspace structures by low-rank representation'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35:1, 2013, pp. 171–184. DOI: 10.1109/TPAMI.2012.88.

251. S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci. 'Visualizing high-dimensional data: Advances in the past decade'. In: *Eurographics Conference on Visualization: State-of-the-Art Reports* (STARs). Ed. by R. Borgo, F. Ganovelli, and I. Viola. The Eurographics Association, 2015. DOI: 10.2312/eurovisstar.20151115.

252. S. Liu, B. Wang, J. J. Thiagarajan, P.-T. Bremer, and V. Pascucci. 'Visual exploration of high-dimensional data through subspace analysis and dynamic projections'. *Computer Graphics Forum* 34:3, 2015, pp. 271–280. DOI: 10.1111/cgf.12639.

253. P. Y. Lum, G. Singh, A. Lehman, T. Ishkanov, M. Vejdemo-Johansson, M. Alagappan, J. Carlsson, and G. Carlsson. 'Extracting insights from the shape of complex data using topology'. *Scientific Reports* 3, 2013, pp. 1–8. DOI: 10.1038/srep01236.

254. P. Y. Lum, A. Lehman, G. Singh, T. Ishkanov, G. Carlsson, and M. Vejdemo-Johansson. 'The topology of politics: Voting connectivity in the U. S. House of Representatives'. In: *NIPS Workshop on Algebraic Topology and Machine Learning*. 2012.

255. L. J. van der Maaten. 'Accelerating t-SNE using tree-based algorithms'. *Journal of Machine Learning Research* 15, 2014, pp. 3221–3245.

256. L. J. van der Maaten and G. E. Hinton. 'Visualizing data using t-SNE'. *Journal of Machine Learning Research* 9:85, 2008, pp. 2579–2605.

257. L. J. van der Maaten, E. O. Postma, and H. J. van den Herik. *Dimensionality reduction: A comparative review*. Technical report 2009-005. Tilburg University, 2009.

258. P. C. Mahalanobis. 'On the generalized distance in statistics'. *Proceedings of the Indian National Science Academy* 2:1, 1936, pp. 49–55.

259. H. Mara. 'Multi-scale integral invariants for robust character extraction from irregular polygon mesh data'. PhD thesis. Ruprecht-Karls-Universität Heidelberg, 2012.

260. H. Mara, S. Krömker, S. Jakob, and B. Breuckmann. 'GigaMesh and Gilgamesh—3D multiscale integral invariant cuneiform character extraction'. In: *Proceedings of the 11th International Symposium on Virtual Reality, Archæology and Intelligent Cultural Heritage* (VAST). Ed. by A. Artusi, M. Joly, G. Lucet, D. Pitzalis, and A. Ribes. The Eurographics Association, 2010, pp. 131–138. DOI: 10.2312/VAST/VAST10/131-138.

261. K. V. Mardia, J. T. Kent, and J. M. Biby. *Multivariate analysis*. Academic Press, London, England, 1979.

262. K. Matković, D. Gračanin, M. Jelović, and H. Hauser. 'Interactive visual steering — Rapid visual prototyping of a common rail injection system'. *IEEE Transactions on Visualization and Computer Graphics* 14:6, 2008, pp. 1699–1706. DOI: 10.1109/TVCG.2008.145.

263. J. A. McFadden. 'The entropy of a point process'. *Journal of the Society for Industrial and Applied Mathematics* 13:4, 1965, pp. 988–994. DOI: 10.1137/0113066.

264. M. Meilă. 'Comparing clusterings—an information based distance'. *Journal of Multivariate Analysis* 98:5, 2007, pp. 873–895. DOI: 10.1016/j.jmva.2006.11.013.

265. F. Mémoli. 'On the use of Gromov–Hausdorff distances for shape comparison'. In: *Eurographics Symposium on Point-Based Graphics*. Ed. by M. Botsch, R. Pajarola, B. Chen, and M. Zwicker. The Eurographics Association, 2007, pp. 81–90. DOI: 10.2312/SPBG/SPBG07/081-090.

266. F. Mémoli and G. Sapiro. 'A theoretical and computational framework for isometry invariant recognition of point cloud data'. *Foundations of Computational Mathematics* 5:3, 2005, pp. 313–347. DOI: 10.1007/s10208-004-0145-y.

267. F. Mémoli and G. Sapiro. 'Distance functions and geodesics on submanifolds of $\mathbb{R}^d$ and point clouds'. *SIAM Journal on Applied Mathematics* 65:4, 2005, pp. 1227–1260. DOI: 10.1137/S003613990342877X.

268. Q. Mérigot, M. Ovsjanikov, and L. J. Guibas. 'Voronoi-based curvature and feature estimation from point clouds'. *IEEE Transactions on Visualization and Computer Graphics* 17:6, 2011, pp. 743–756. DOI: 10.1109/TVCG.2010.261.

269. M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. 'Discrete differential-geometry operators for triangulated 2-manifolds'. In: *Visualization and Mathematics III*. Ed. by H.-C. Hege and K. Polthier. Springer, Heidelberg, Germany, 2003, pp. 35–57. DOI: 10.1007/978-3-662-05105-4_2.

270. J. Milnor. *Morse theory*. Princeton University Press, Princeton, NJ, USA, 1963.

271. B. Mokbel, W. Lueks, A. Gisbrecht, and B. Hammer. 'Visualizing the quality of dimensionality reduction'. *Neurocomputing* 112, 2013, pp. 109–123. DOI: 10.1016/j.neucom.2012.11.046.

272. D. Morozov. 'Homological illusions of persistence and stability'. PhD thesis. Duke University, 2008.

273. M. Morse. *The calculus of variations in the large*. Colloquium Publications 18. American Mathematical Society, Providence, RI, USA, 1934.

274. T. Mühlbacher and H. Piringer. 'A partition-based framework for building and validating regression models'. *IEEE Transactions on Visualization and Computer Graphics* 19:12, 2013, pp. 1962–1971. DOI: 10.1109/TVCG.2013.125.

275. M. Muja and D. G. Lowe. 'Scalable nearest neighbor algorithms for high dimensional data'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36:11, 2014, pp. 2227–2240. DOI: 10.1109/TPAMI.2014.2321376.

276. E. Munch. 'Applications of persistent homology to time varying systems'. PhD thesis. Duke University, 2013.

277. J. R. Munkres. *Elements of algebraic topology*. Addison–Wesley Publishing Company, Inc., Reading, MA, USA, 1984.

278. T. Munzner. 'H3: Laying out large directed graphs in 3D hyperbolic space'. In: *IEEE Symposium on Information Visualization*. Ed. by J. Dill and N. Gershon. IEEE Computer Society Press, Los Alamitos, CA, USA, 1997, pp. 2–10. DOI: 10.1109/INFVIS.1997.636718.

279. E. A. Nadaraya. 'On estimating regression'. *Theory of Probability & Its Applications* 9:1, 1964, pp. 141–142. DOI: 10.1137/1109020.

280. E. J. Nam, Y. Han, K. M. A. Zelenyuk, and D. Imre. 'ClusterSculptor: A visual analytics tool for high-dimensional data'. In: *IEEE Symposium on Visual Analytics Science and Technology* (VAST). Ed. by W. Ribarsky and J. Dill. Curran Associates, Inc., Red Hook, NY, USA, 2007, pp. 75–82. DOI: 10.1109/VAST.2007.4388999.

281. J. E. Nam and K. Mueller. 'TripAdvisor^{N-D}: A tourism-inspired high-dimensional space exploration framework with overview and detail'. *IEEE Transactions on Visualization and Computer Graphics* 19:2, 2013, pp. 291–305. DOI: 10.1109/TVCG.2012.65.

282. H. Narayanan and S. Mitter. 'Sample complexity of testing the manifold hypothesis'. In: *Advances in Neural Information Processing Systems 23* (NIPS). Ed. by J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta. Curran Associates, Inc., Red Hook, NY, USA, 2010, pp. 1786–1794.

283. M. Nicolau, A. J. Levine, and G. Carlsson. '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* (PNAS) 108:17, 2011, pp. 7265–7270. DOI: 10.1073/pnas.1102826108.

284. J. Nilsson, T. Fioretos, M. Höglund, and M. Fontes. 'Approximate geodesic distances reveal biologically relevant structures in microarray data'. *Bioinformatics* 20:6, 2004, pp. 874–880. DOI: 10.1093/bioinformatics/btg496.

285. P. Niyogi, S. Smale, and S. Weinberger. 'Finding the homology of submanifolds with high confidence from random samples'. *Discrete & Computational Geometry* 39:1, 2008, pp. 419–441. DOI: 10.1007/s00454-008-9053-2.

286. P. Oesterling, C. Heine, H. Jänicke, G. Scheuermann, and G. Heyer. 'Visualization of high-dimensional point clouds using their density distribution's topology'. *IEEE Transactions on Visualization and Computer Graphics* 17:11, 2011, pp. 1547–1559. DOI: 10.1109/TVCG.2011.27.

287. P. Oesterling, C. Heine, G. H. Weber, and G. Scheuermann. 'Visualizing *n*D point clouds as topological landscape profiles to guide local data analysis'. *IEEE Transactions on Visualization and Computer Graphics* 19:3, 2013, pp. 514–526. DOI: 10.1109/TVCG.2012.120.

288. V. Pascucci, K. Cole-McLaughlin, and G. Scorzelli. 'The Toporrery: Computation and presentation of multiresolution topology'. In: *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*. Ed. by T. Möller, B. Hamann, and R. D. Russell. Springer, Heidelberg, Germany, 2009, pp. 19–40. DOI: 10.1007/b106657_2.

289. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. 'Scikit-learn: Machine Learning in Python'. *Journal of Machine Learning Research* 12, 2011, pp. 2825–2830.

290. W. Peng, M. O. Ward, and E. A. Rundensteiner. 'Clutter reduction in multi-dimensional data visualization using dimension reordering'. In: *IEEE Symposium on Information Visualization*. Ed. by M. Ward and T. Munzner. IEEE Computer Society Press, Los Alamitos, CA, USA, 2004, pp. 89–96. DOI: 10.1109/INFVIS.2004.15.

291. J. A. Perea, A. Deckard, S. B. Haase, and J. Harer. 'SW1PerS: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data'. *BMC Bioinformatics* 16:1, 2015, pp. 1–12. DOI: 10.1186/s12859-015-0645-6.

292. J. A. Perea and J. Harer. 'Sliding windows and persistence: An application of topological methods to signal analysis'. *Foundations of Computational Mathematics* 15:3, 2014, pp. 799–838. DOI: 10.1007/s10208-014-9206-z.

293. R. M. Pickett and G. G. Grinstein. 'Iconographic displays for visualizing multidimensional data'. In: *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*. Vol. 1. International Academic Publishers, Beijing, China, 1988, pp. 514–519. DOI: 10.1109/ICSMC.1988.754351.

294. A. Pilhöfer, A. Gribov, and A. Unwin. 'Comparing clusterings using Bertin's idea'. *IEEE Transactions on Visualization and Computer Graphics* 18:12, 2012, pp. 2506–2515. DOI: 10.1109/TVCG.2012.207.

295. G. Plonka and Y. Zheng. 'Relation between total variation and persistence distance and its application in signal processing'. *Advances in Computational Mathematics* 42:3, 2016, pp. 651–674. DOI: 10.1007/s10444-015-9438-8.

296. K. T. Poole and H. L. Rosenthal. *Ideology and Congress*. 2nd ed. Transaction Publishers, Piscataway, NJ, USA, 2007.

297. M. A. Porter, P. J. Mucha, M. E. J. Newman, and C. M. Warmbrand. 'A network analysis of committees in the U. S. House of Representatives'. *Proceedings of the National Academy of Sciences of the United States of America* (PNAS) 102:20, 2005, pp. 7057–7062. DOI: 10.1073/pnas.0500191102.

298. H. Pottmann, J. Wallner, Q.-X. Huang, and Y.-L. Yang. 'Integral invariants for robust geometry processing'. *Computer Aided Geometric Design* 26:1, 2009, pp. 37–60. DOI: 10.1016/j.cagd.2008.01.002.

299. H. Pottmann, J. Wallner, Y.-L. Yang, Y.-K. Lai, and S.-M. Hu. 'Principal curvatures from the integral invariant viewpoint'. *Computer Aided Geometric Design* 24:8–9, 2007, pp. 428–442. DOI: 10.1016/j.cagd.2007.07.004.

300. H. C. Purchase, R. F. Cohen, and M. James. 'Validating graph drawing æsthetics'. In: *Graph Drawing*. Ed. by F. J. Brandenburg. Lecture Notes in Computer Science 1027. Springer, Heidelberg, Germany, 1996, pp. 435–446. DOI: 10.1007/BFb0021827.

301. J. R. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers, San Mateo, CA, USA, 1993.

302. P. T. Quinlan and R. N. Wilton. 'Grouping by proximity or similarity? Competition between the gestalt principles in vision'. *Perception* 27:4, 1998, pp. 417–430. DOI: 10.1068/p270417.

303. W. M. Rand. 'Objective criteria for the evaluation of clustering methods'. *Journal of the American Statistical Association* 66:336, 1971, pp. 846–850. DOI: 10.1080/01621459.1971.10482356.

304. C. K. Reddy and B. Vinzamuri. 'A survey of partitional and hierarchical clustering algorithms'. In: *Data clustering. Algorithms and applications*. Ed. by C. C. Aggarwal and C. K. Reddy. Data Mining and Knowledge Discovery. Chapman & Hall/CRC, Boca Raton, FL, USA, 2014. Chap. 4, pp. 87–110.

305. D. Reem. 'The geometric stability of Voronoi diagrams with respect to small changes of the sites'. In: *Proceedings of the 27th Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 2011, pp. 254–263. DOI: 10.1145/1998196.1998234.

306. J. Reininghaus, S. Huber, U. Bauer, and R. Kwitt. 'A stable multi-scale kernel for topological machine learning'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR). Curran Associates, Inc., Red Hook, NY, USA, 2015, pp. 4741–4748. DOI: 10.1109/CVPR.2015.7299106.

307. M. Reuter. 'Hierarchical shape segmentation and registration via topological features of Laplace–Beltrami eigenfunctions'. *International Journal of Computer Vision* 89:2, 2009, pp. 287–308. DOI: 10.1007/s11263-009-0278-1.

308. M. Reuter, S. Biasotti, D. Giorgi, G. Patanè, and M. Spagnuolo. 'Discrete Laplace–Beltrami operators for shape analysis and segmentation'. *Computers & Graphics* 33:3, 2009, pp. 381–390. DOI: 10.1016/j.cag.2009.03.005.

309. P. Rheingans and M. desJardins. 'Visualizing high-dimensional predicitive model quality'. In: *Proceedings of the 11th Annual Conference on Visualization*. Ed. by T. Ertl, B. Hamann, and A. Varshney. IEEE Computer Society Press, Los Alamitos, CA, USA, 2000, pp. 493–496. DOI: 10.1109/VISUAL.2000.885740.

310. B. Rieck and H. Leitte. 'Agreement analysis of quality measures for dimensionality reduction'. In: *Workshop on Topology-Based Methods in Visualization* (TopoInVis). To appear in *Topological Methods in Data Analysis and Visualization IV*. Annweiler, Germany, 2015.

311. B. Rieck and H. Leitte. 'Comparing dimensionality reduction methods using data descriptor landscapes'. In: *Symposium on Visualization in Data Science* (VDS) *at IEEE VIS*. Chicago, IL, USA, 2015.

312. B. Rieck and H. Leitte. '*Shall I compare thee to a network?* — Visualizing the topological structure of Shakespeare's plays'. In: *Workshop on Visualization for the Digital Humanities at IEEE VIS*. Baltimore, MD, USA, 2016.

313. B. Rieck and H. Leitte. 'Enhancing comparative model analysis using persistent homology'. In: *Workshop on Visualization for Predictive Analytics at IEEE VIS*. Paris, France, 2014.

314. B. Rieck and H. Leitte. 'Exploring and comparing clusterings of multivariate data sets using persistent homology'. *Computer Graphics Forum* 35:3, 2016, pp. 81–90. DOI: 10.1111/cgf.12884.

315. B. Rieck and H. Leitte. 'Persistent homology for the evaluation of dimensionality reduction schemes'. *Computer Graphics Forum* 34:3, 2015, pp. 431–440. DOI: 10.1111/cgf.12655.

316. B. Rieck and H. Leitte. 'Structural analysis of multivariate point clouds using simplicial chains'. *Computer Graphics Forum* 33:8, 2014, pp. 28–37. DOI: 10.1111/cgf.12398.

317. B. Rieck, H. Mara, and S. Krömker. 'Unwrapping highly-detailed 3D meshes of rotationally symmetric man-made objects'. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* II-5/W1, 2013, pp. 259–264. DOI: 10.5194/isprsannals-II-5-W1-259-2013.

318. B. Rieck, H. Mara, and H. Leitte. 'Multivariate data analysis using persistence-based filtering and topological signatures'. *IEEE Transactions on Visualization and Computer Graphics* 18:12, 2012, pp. 2382–2391. DOI: 10.1109/TVCG.2012.248.

319. V. Robins. 'Towards computing homology from finite approximations'. *Topology Proceedings* 24, 1999, pp. 503–532.

320. P. J. Rousseeuw. 'Silhouettes: A graphical aid to the interpretation and validation of cluster analysis'. *Journal of Computational and Applied Mathematics* 20, 1987, pp. 53–65. DOI: 10.1016/0377-0427(87)90125-7.

321. P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*. John Wiley & Sons, Inc., New York, NY, USA, 2003. DOI: 10.1002/0471725382.

322. S. T. Roweis and L. K. Saul. 'Nonlinear dimensionality reduction by locally linear embedding'. *Science* 290:5500, 2000, pp. 2323–2326. DOI: 10.1126/science.290.5500.2323.

323. N. Sauber, H. Theisel, and H.-P. Seidel. 'Multifield-graphs: An approach to visualizing correlations in multifield scalar data'. *IEEE Transactions on Visualization and Computer Graphics* 12:5, 2006, pp. 917–924. DOI: 10.1109/TVCG.2006.165.

324. L. K. Saul and S. T. Roweis. 'Think globally, fit locally: Unsupervised learning of low dimensional manifolds'. *Journal of Machine Learning Research* 4, 2003, pp. 119–155.

325. D. Schneider, C. Heine, H. Carr, and G. Scheuermann. 'Interactive comparison of multifield scalar data based on largest contours'. *Computer Aided Geometric Design* 30:6, 2013, pp. 521–528. DOI: 10.1016/j.cagd.2012.03.023.

326. D. Schneider, A. Wiebel, H. Carr, M. Hlawitschka, and G. Scheuermann. 'Interactive comparison of scalar fields based on largest contours with applications to flow visualization'. *IEEE Transactions on Visualization and Computer Graphics* 14:6, 2008, pp. 1475–1482. DOI: 10.1109/TVCG.2008.143.

327. T. Schreck, J. Bernard, T. von Landesberger, and J. Kohlhammer. 'Visual cluster analysis of trajectory data with interactive Kohonen maps'. *Information Visualization* 8:1, 2009, pp. 14–29. DOI: 10.1057/ivs.2008.29.

328. D. Sculley. 'Web-scale k-means clustering'. In: *Proceedings of the 19th International World Wide Web Conference* (WWW). ACM Press, New York, NY, USA, 2010, pp. 1177–1178. DOI: 10.1145/1772690.1772862.

329. M. Sedlmair, M. Brehmer, S. Ingram, and T. Munzner. *Dimensionality reduction in the wild: Gaps and guidance*. Technical report 2012-03. University of British Columbia, 2012.

330. M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. 'Visual parameter space analysis: A conceptual framework'. *IEEE Transactions on Visualization and Computer Graphics* 20:12, 2014, pp. 2161–2170. DOI: 10.1109/TVCG.2014.2346321.

331. M. Sedlmair, T. Munzner, and M. Tory. 'Empirical guidance on scatterplot and dimension reduction technique choices'. *IEEE Transactions on Visualization and Computer Graphics* 19:12, 2013, pp. 2634–2643. DOI: 10.1109/TVCG.2013.153.

332. S. Segarra and A. Ribeiro. 'Stability and continuity of centrality measures in weighted graphs'. *IEEE Transactions on Signal Processing* 64:3, 2016, pp. 543–555. DOI: 10.1109/TSP.2015.2486740.

333. J. Seo and B. Shneiderman. 'Interactively exploring hierarchical clustering results'. *Computer* 35:7, 2002, pp. 80–86. DOI: 10.1109/MC.2002.1016905.

334. D. R. Sheehy. 'Linear-size approximations to the Vietoris–Rips filtration'. *Discrete & Computational Geometry* 49:4, 2013, pp. 778–796. DOI: 10.1007/s00454-013-9513-1.

335. B. Shneiderman. 'The eyes have it: A task by data type taxonomy for information visualizations'. In: *IEEE Symposium on Visual Languages*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1996, pp. 336–343. DOI: 10.1109/VL.1996.545307.

336. B. Shneiderman. 'Tree visualization with tree-maps: 2D space-filling approach'. *ACM Transactions on Graphics* 11:1, 1992, pp. 92–99. DOI: 10.1145/102377.115768.

337. V. de Silva and G. Carlsson. 'Topological estimation using witness complexes'. In: *Symposium on Point-Based Graphics*. Ed. by M. Gross, H. Pfister, M. Alexa, and S. Rusinkiewicz. The Eurographics Association, 2004. DOI: 10.2312/SPBG/SPBG04/157-166.

338. V. de Silva and R. Ghrist. 'Coverage in sensor networks via persistent homology'. *Algebraic & Geometric Topology* 7:1, 2007, pp. 339–358. DOI: 10.2140/agt.2007.7.339.

339. V. de Silva, P. Skraba, and M. Vejdemo-Johansson. 'Topological analysis of recurrent systems'. In: *NIPS Workshop on Algebraic Topology and Machine Learning*. 2012. URL: http://www.cs.cmu.edu/~sbalakri/Topology_final_versions/dS-S-VJ-final.pdf.

340. V. de Silva and J. B. Tenenbaum. 'Global versus local methods in nonlinear dimensionality reduction'. In: *Advances in Neural Information Processing Systems 15* (NIPS). Ed. by S. Becker, S. Thrun, and K. Obermayer. MIT Press, Cambridge, MA, USA, 2003, pp. 705–712.

341. B. W. Silverman. *Density estimation for statistics and data analysis*. Monographs on Statistics and Applied Probability 26. Chapman & Hall/CRC, Boca Raton, FL, USA, 1986.

342. G. Singh, F. Mémoli, and G. Carlsson. 'Topological methods for the analysis of high dimensional data sets and 3D object recognition'. In: *Eurographics Symposium on Point-Based Graphics*. Ed. by M. Botsch, R. Pajarola, B. Chen, and M. Zwicker. The Eurographics Association, 2007. DOI: 10.2312/SPBG/SPBG07/091-100.

343. J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. 'Discovering objects and their location in images'. In: *Proceedings of the 10th IEEE International Conference on Computer Vision* (ICCV). Vol. 1. IEEE Computer Society Press, Los Alamitos, CA, USA, 2005, pp. 370–377. DOI: 10.1109/ICCV.2005.77.

344. S. Smale. 'On gradient dynamical systems'. *Annals of Mathematics* 74:1, 1961, pp. 199–206. DOI: 10.2307/1970311.

345. P. H. A. Sneath. 'The application of computers to taxonomy'. *Microbiology* 17:1, 1957, pp. 201–226. DOI: 10.1099/00221287-17-1-201.

346. B. Sohn and C. Bajaj. 'Time-varying contour topology'. *IEEE Transactions on Visualization and Computer Graphics* 12:1, 2006, pp. 14–25. DOI: 10.1109/TVCG.2006.16.

347. R. R. Sokal and F. J. Rohlf. 'The comparison of dendrograms by objective methods'. *Taxon* 11:2, 1962, pp. 33–40. DOI: 10.2307/1217208.

348. B. Spence, L. Tweedie, H. Dawkes, and H. Su. 'Visualisation for functional design'. In: *IEEE Symposium on Information Visualization*. Ed. by N. Gershon and S. Eick. IEEE Computer Society Press, Los Alamitos, CA, USA, 1995, pp. 4–10. DOI: 10.1109/INFVIS.1995.528680.

349. J. Steele and N. Iliinsky, eds. *Beautiful visualization. Looking at data through the eyes of experts*. O'Reilly, Sebastopol, CA, USA, 2010.

350. S. S. Stevens. 'On the theory of scales of measurement'. *Science* 103:2684, 1946, pp. 677–680. DOI: 10.1126/science.103.2684.677.

351. S. S. Stevens. *Psychophysics. Introduction to its perceptual, neural, and social prospects*. Transaction Publishers, Piscataway, NJ, USA, 1986.

352. J. Sun, M. Ovsjanikov, and L. J. Guibas. 'A concise and provably informative multi-scale signature based on heat diffusion'. *Computer Graphics Forum* 28:5, 2009, pp. 1383–1392. DOI: 10.1111/j.1467-8659.2009.01515.x.

353. G. J. Székely and M. L. Rizzo. 'Energy statistics: A class of statistics based on distances'. *Journal of Statistical Planning and Inference* 143:8, 2013, pp. 1249–1272. DOI: 10.1016/j.jspi.2013.03.018.

354. P. Tan, M. Steinbach, and V. Kumar. *Introduction to data mining*. 2nd ed. Addison–Wesley Publishing Company, Inc., Boston, MA, USA, 2006.

355. R. E. Tarjan. 'Efficiency of a good but not linear set union algorithm'. *Journal of the ACM* 22:2, 1975, pp. 215–225. DOI: 10.1145/321879.321884.

356. A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H. Theisel, M. Magnor, and D. A. Keim. 'Combining automated analysis and visualization techniques for effective exploration of high-dimensional data'. In: *IEEE Symposium on Visual Analytics Science and Technology* (VAST). Ed. by J. Stasko and J. J. van Wijk. Curran Associates, Inc., Red Hook, NY, USA, 2009, pp. 59–66. DOI: 10.1109/VAST.2009.5332628.

357. A. Tatu, P. Bak, E. Bertini, D. Keim, and J. Schneidewind. 'Visual quality metrics and human perception: An initial study on 2D projections of large multidimensional data'. In: *Proceedings of the International Conference on Advanced Visual Interfaces* (AVI). ACM Press, New York, NY, USA, 2010, pp. 49–56. DOI: 10.1145/1842993.1843002.

358. A. Tatu, F. Maaß, I. Färber, E. Bertini, T. Schreck, T. Seidl, and D. Keim. 'Subspace search and visualization to make sense of alternative clusterings in high-dimensional data'. In: *IEEE Symposium on Visual Analytics Science and Technology* (VAST). Ed. by G. Santucci and M. O. Ward. Curran Associates, Inc., Red Hook, NY, USA, 2012, pp. 63–72. DOI: 10.1109/VAST.2012.6400488.

359. J. B. Tenenbaum, V. de Silva, and J. C. Langford. 'A global geometric framework for nonlinear dimensionality reduction'. *Science* 290:5500, 2000, pp. 2319–2323. DOI: 10.1126/science.290.5500.2319.

360. I. V. Tetko, V. Y. Tanchuk, T. N. Kasheva, and A. E. Villa. 'Estimation of aqueous solubility of chemical compounds using E-state indices'. *Journal of Chemical Information and Computer Sciences* 41:6, 2001, pp. 1488–1493. DOI: 10.1021/ci000392t.

361. J. Tierny. 'Reeb graph based 3D shape modeling and applications'. PhD thesis. Université des Sciences et Technologies de Lille, 2008.

362. J. Tierny and V. Pascucci. 'Generalized topological simplification of scalar fields on surfaces'. *IEEE Transactions on Visualization and Computer Graphics* 18:12, 2012, pp. 2005–2013. DOI: 10.1109/TVCG.2012.228.

363. C. Tofallis. 'A better measure of relative prediction accuracy for model selection and model estimation'. *Journal of the Operational Research Society* 66:8, 2015, pp. 1352–1362. DOI: 10.1057/jors.2014.103.

364. G. Toussaint. 'Geometric proximity graphs for improving nearest neighbor methods in instance-based learning and data mining'. *International Journal of Computational Geometry & Applications* 15:2, 2005, pp. 101–150. DOI: 10.1142/S0218195905001622.

365. A. Treisman. 'Preattentive processing in vision'. *Computer Vision, Graphics, and Image Processing* 31:2, 1985, pp. 156–177. DOI: 10.1016/S0734-189X(85)80004-9.

366. K. E. Trenberth. 'The definition of El Niño'. *Bulletin of the American Meteorological Society* 78:12, 1997, pp. 2771–2777. DOI: 10.1175/1520-0477(1997)078<2771:TDOENO>2.0.CO;2.

367. E. R. Tufte. *Envisioning information*. Graphics Press, Cheshire, CT, USA, 1990.

368. E. R. Tufte. *The visual display of quantitative information*. 2nd ed. Graphics Press, Cheshire, CT, USA, 2001.

369. J. W. Tukey. *Exploratory data analysis*. Addison–Wesley Publishing Company, Inc., Reading, MA, USA, 1977.

370. J. W. Tukey. 'The future of data analysis'. *The Annals of Mathematical Statistics* 33:1, 1962, pp. 1–67. DOI: 10.1214/aoms/1177704711.

371. J. W. Tukey. 'We need both exploratory and confirmatory'. *The American Statistician* 34:1, 1980, pp. 23–25. DOI: 10.1080/00031305.1980.10482706.

372. A. Unger, S. Schulte, V. Klemann, and D. Dransch. 'A visual analysis concept for the validation of geoscientific simulation models'. *IEEE Transactions on Visualization and Computer Graphics* 18:12, 2012, pp. 2216–2225. DOI: 10.1109/TVCG.2012.190.

373. S. van Dongen. 'Graph clustering by flow simulation'. PhD thesis. Utrecht University, 2000.

374. P. J. Verveer and R. P. Duin. 'An evaluation of intrinsic dimensionality estimators'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17:1, 1995, pp. 81–86. DOI: 10.1109/34.368147.

375. L. Vietoris. 'Über den höheren Zusammenhang kompakter Räume und eine Klasse von zusammenhangstreuen Abbildungen'. *Mathematische Annalen* 97:1, 1927, pp. 454–472.

376. C. Villani. *Optimal transport. Old and new*. Grundlehren der mathematischen Wissenschaften 338. Springer, Heidelberg, Germany, 2009. DOI: 10.1007/978-3-540-71050-9.

377. K. Volkert. 'Poincaré's homology sphere'. *Bulletin of the Manifold Atlas*, 2013. URL: http://www.boma.mpim-bonn.mpg.de/data/40print.pdf.

378. J. Wagemans, J. H. Elder, M. Kubovy, S. E. Palmer, M. A. Peterson, M. Singh, and R. von der Heydt. 'A century of gestalt psychology in visual perception: I. Perceptual grouping and figure-ground organization'. *Psychological Bulletin* 138:6, 2012, pp. 1172–1217. DOI: 10.1037/a0029333.

379. J. Wagemans, J. Feldman, S. Gepshtein, R. Kimchi, J. R. Pomerantz, P. A. van der Helm, and C. van Leeuwen. 'A century of gestalt psychology in visual perception: II. Conceptual and theoretical foundations'. *Psychological Bulletin* 138:6, 2012, pp. 1218–1252. DOI: 10.1037/a0029334.

380. B. Wang and G. Wei. 'Object-oriented persistent homology'. *Journal of Computational Physics* 305, 2016, pp. 276–299. DOI: 10.1016/j.jcp.2015.10.036.

381. B. Wang, B. Summa, V. Pascucci, and M. Vejdemo-Johansson. 'Branching and circular features in high dimensional data'. *IEEE Transactions on Visualization and Computer Graphics* 17:12, 2011, pp. 1902–1911. DOI: 10.1109/TVCG.2011.177.

382. C. Wang, H. Yu, and K. Ma. 'Importance-driven time-varying data visualization'. *IEEE Transactions on Visualization and Computer Graphics* 14:6, 2008, pp. 1547–1554. DOI: 10.1109/TVCG.2008.140.

383. M. O. Ward. 'A taxonomy of glyph placement strategies for multidimensional data visualization'. *Information Visualization* 1:3–4, 2002, pp. 194–210. DOI: 10.1057/palgrave.ivs.9500025.

384. M. O. Ward. 'Multivariate data glyphs: Principles and practice'. In: *Handbook of Data Visualization*. Ed. by C. Chen, W. Härdle, and A. Unwin. Springer, Heidelberg, Germany, 2008, pp. 179–198. DOI: 10.1007/978-3-540-33037-0_8.

385. F. W. Warner. *Foundations of differentiable manifolds and Lie Groups*. Graduate Texts in Mathematics 94. Springer, New York, NY, USA, 1983. DOI: 10.1007/978-1-4757-1799-0.

386. L. Wasserman. *All of statistics. A concise course in statistical inference*. Springer, New York, NY, USA, 2004. DOI: 10.1007/978-0-387-21736-9.

387. G. H. Weber, P.-T. Bremer, M. Day, J. Bell, and V. Pascucci. 'Feature tracking using Reeb graphs'. In: *Topological Methods in Data Analysis and Visualization. Theory, Algorithms, and Applications*. Ed. by V. Pascucci, X. Tricoche, H. Hagen, and J. Tierny. Springer, Heidelberg, Germany, 2011, pp. 241–253. DOI: 10.1007/978-3-642-15014-2_20.

388. G. H. Weber, P.-T. Bremer, and V. Pascucci. 'Topological landscapes: A terrain metaphor for scientific data'. *IEEE Transactions on Visualization and Computer Graphics* 13:6, 2007, pp. 1416–1423. DOI: 10.1109/TVCG.2007.70601.

389. J. N. Weinstein. 'A postgenomic visual icon'. *Science* 319:5871, 2008, pp. 1772–1773. DOI: 10.1126/science.1151888.

390. J. N. Weinstein, T. G. Myers, P. M. O'Connor, S. H. Friend, A. J. Fornace Jr., K. W. Kohn, T. Fojo, S. E. Bates, L. V. Rubinstein, N. L. Anderson, J. K. Buolamwini, W. W. van Osdol, A. P. Monks, D. A. Scudiero, E. A. Sausville, D. W. Zaharevitz, B. Bunow, V. N. Viswanadhan, G. S. Johnson, R. E. Wittes, and K. D. Paull. 'An information-intensive approach to the molecular pharmacology of cancer'. *Science* 275:5298, 1997, pp. 343–349. DOI: 10.1126/science.275.5298.343.

391. J. H. C. Whitehead. 'On C^1-complexes'. *Annals of Mathematics* 41:4, 1940, pp. 809–824. DOI: 10.2307/1968861.

392. L. Wilkinson and G. Wills. 'Scagnostics distributions'. *Journal of Computational and Graphical Statistics* 17:2, 2008, pp. 473–491. DOI: 10.1198/106186008X320465.

393. C. Woods, E. Teeter, and G. Emberling, eds. *Visible language. Inventions of writing in the ancient Middle East and beyond*. Oriental Institute Museum Publications 32. The Oriental Institute of the University of Chicago, Chicago, IL, USA, 2010.

394. X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg. 'Top 10 algorithms in data mining'. *Knowledge and Information Systems* 14:1, 2008, pp. 1–37. DOI: 10.1007/s10115-007-0114-2.

395. K. Xia and G. Wei. 'Multidimensional persistence in biomolecular data'. *Journal of Computational Chemistry* 36:20, 2015, pp. 1502–1520. DOI: 10.1002/jcc.23953.

396. H. Xiong and Z. Li. 'Clustering validation measures'. In: *Data clustering. Algorithms and applications*. Ed. by C. C. Aggarwal and C. K. Reddy. Data Mining and Knowledge Discovery. Chapman & Hall/CRC, Boca Raton, FL, USA, 2014. Chap. 23, pp. 571–605.

397. D. Xu and Y. Tian. 'A comprehensive survey of clustering algorithms'. *Annals of Data Science* 2:2, 2015, pp. 165–193. DOI: 10.1007/s40745-015-0040-1.

398. J. Yang, W. Peng, M. O. Ward, and E. A. Rundensteiner. 'Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets'. In: *IEEE Symposium on Information Visualization*. Ed. by T. Munzner and S. North. IEEE Computer Society Press, Los Alamitos, CA, USA, 2003, pp. 105–112. DOI: 10.1109/INFVIS.2003.1249015.

399. Y.-L. Yang, Y.-K. Lai, S.-M. Hu, and H. Pottmann. 'Robust principal curvatures on multiple scales'. In: *Eurographics Symposium on Geometry Processing*. Ed. by A. Sheffer and K. Polthier. The Eurographics Association, 2006, pp. 223–226. DOI: 10.2312/SGP/SGP06/223-226.

400. Y. Yao, J. Sun, X. Huang, G. R. Bowman, G. Singh, M. Lesnick, L. J. Guibas, V. S. Pande, and G. Carlsson. 'Topological methods for exploring low-density states in biomolecular folding pathways'. *The Journal of Chemical Physics* 130:14, 2009. DOI: 10.1063/1.3103496.

401. I.-C. Yeh. 'Modeling of strength of high-performance concrete using artificial neural networks'. *Cement and Concrete Research* 28:12, 1998, pp. 1797–1808. DOI: 10.1016/S0008-8846(98)00165-3.

402. M. J. Zaki and W. Meira Jr. *Data mining and analysis. Fundamental concepts and algorithms*. Cambridge University Press, Cambridge, England, 2014.

403. R. S. Zemel and M. A. Carreira-Perpiñán. 'Proximity graphs for clustering and manifold learning'. In: *Advances in Neural Information Processing Systems 17* (NIPS). Ed. by L. K. Saul, Y. Weiss, and L. Bottou. MIT Press, Cambridge, MA, USA, 2004, pp. 225–232.

404. Y. Zhang, W. Luo, E. A. Mack, and R. Maciejewski. 'Visualizing the impact of geographical variations on multivariate clustering'. *Computer Graphics Forum* 35:3, 2016, pp. 101–110. DOI: 10.1111/cgf.12886.

405. Z. Zhang and H. Zha. 'Principal manifolds and nonlinear dimensionality reduction via tangent space alignment'. *SIAM Journal on Scientific Computing* 26:1, 2004, pp. 313–338. DOI: 10.1137/S1064827502419154.

406. A. J. Zomorodian. 'Computing and comprehending topology: Persistence and hierarchical Morse complexes'. PhD thesis. University of Illinois at Urbana–Champaign, 2001.

407. A. J. Zomorodian. 'Fast construction of the Vietoris–Rips complex'. *Computers & Graphics* 34:3, 2010, pp. 263–271. DOI: 10.1016/j.cag.2010.03.007.

408. A. J. Zomorodian. 'The tidy set: A minimal simplicial set for computing homology of clique complexes'. In: *Proceedings of the 26th Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 2010, pp. 257–266. DOI: 10.1145/1810959.1811004.

409. A. J. Zomorodian and G. Carlsson. 'Computing persistent homology'. *Discrete & Computational Geometry* 33:2, 2005, pp. 249–274. DOI: 10.1007/s00454-004-1146-y.

410. A. J. Zomorodian and G. Carlsson. 'Localized homology'. *Computational Geometry* 41:3, 2008, pp. 126–148. DOI: 10.1016/j.comgeo.2008.02.003.