Introduction

A picture being worth a thousand words, let us introduce our subject by showing a toy example coming from data analysis. Consider the data set of Figure 0.1, which is composed of 176 points sampled along 11 congruent letter-B shapes arranged into a letter A in the plane. When asking about the shape represented by this data set, one usually gets the answer: "It depends", followed by a list of possible choices, the most common of which being "eleven B's" and "one A". To these choices one could arguably add a third obvious possibility: "176 points". What differentiates these choices is the scale at which each one of them fits the data.

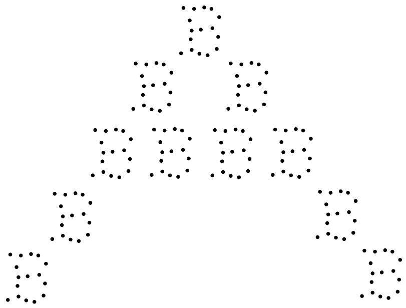

FIGURE 0.1. A planar point set with several underlying geometric structures at different scales.

Finding the 'right' scale(s) at which to process a given data set is a common problem faced across the data analysis literature. Most approaches simply ignore it and delegate the choice of scale to the user, who is then reduced to tuning some parameter blindly, usually by trial-and-error. Sometimes the parameter to tune does not even have a direct interpretation as a scale, which makes things even harder. This is where multiscale approaches distinguish themselves: by processing the data at all scales at once, they do not rely on a particular choice of scale. Their feedback gives the user a precise understanding of the relationship between the choice of input parameter and the output to be expected. Eventually, finding the 'right' scale to be used to produce the final output is still left to the user, however (s)he can now make an informed choice of parameter.

As an illustration, Figure 0.2 shows the result obtained by hierarchical agglomerative clustering on the aforementioned data set. The hierarchy reveals three relevant scales: at low levels (between 0 and 4), the clustering has one cluster per data point; at intermediate levels (between 8 and 12), the clustering has one cluster per letter B; at the highest level (above 16), there is only one cluster left, which spans the entire letter A.

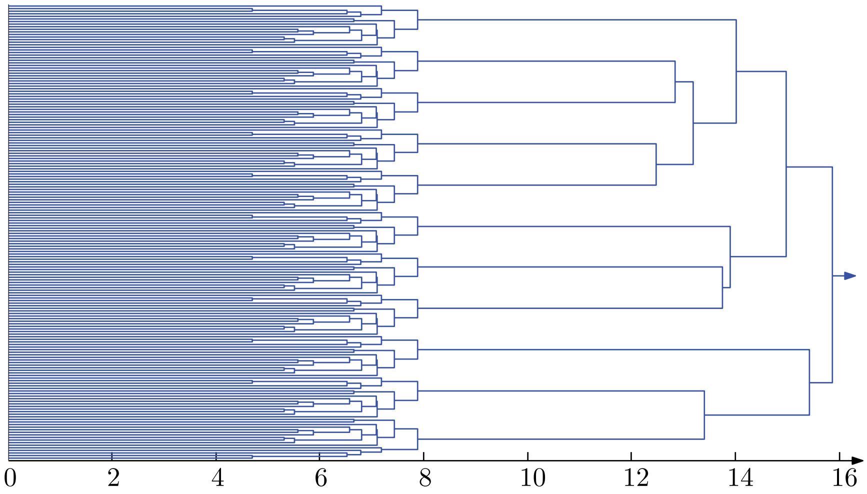

FIGURE 0.2. The hierarchy (also called *dendrogram*) produced by *single-linkage* clustering on the data set of Figure 0.1.

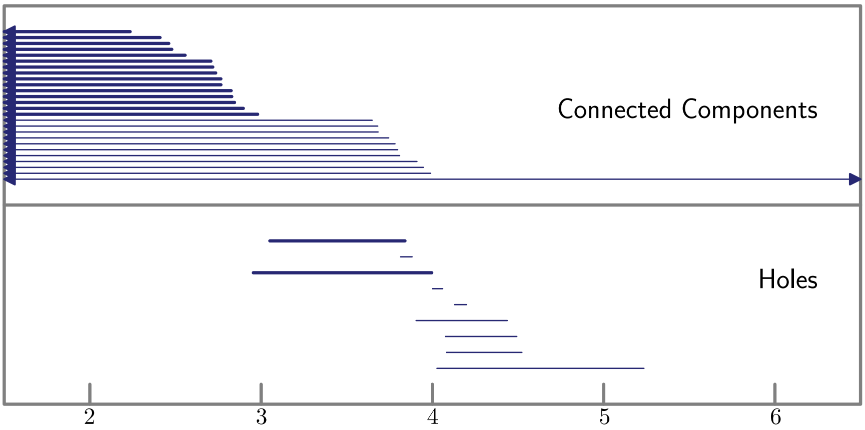

FIGURE 0.3. The barcode produced by persistence on the data set of Figure 0.1. The abscissa line represents the geometric scale with a logarithmic graduation. Left (resp. right) arrows mark left- (resp. right-) infinite intervals, while thin (resp. bold) bars mark intervals with multiplicity 1 (resp. 11).

Persistence produces the same hierarchy but uses a simplified representation for it, shown in the upper half of Figure 0.3. This representation forgets the actual merge pattern between the clusters. When two clusters are merged, they no longer produce a new cluster corresponding to their union in the hierarchy. Instead, one of them ceases to be treated as an independent cluster, to the benefit of the other. The choice of the winner is arbitrary in this case, however in general it is driven by a principle called the *elder rule*, which will be illustrated in the upcoming Figure 0.5. The resulting collection of horizontal bars is called a *persistence barcode*. Each bar is associated to a single data point and represents its *persistence* as an independent

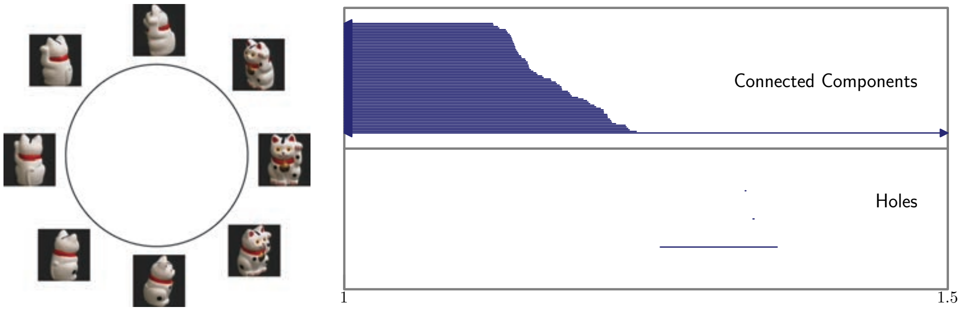

Figure 0.4. Left: a collection of 72 color images of size 128×128 pixels coming from the Columbia Object Image Library [203]. The images were obtained by taking pictures of an object while rotating around it. Each image gives a data point in 49 152 dimensions, where by construction the data set lies near some simple closed curve. Right: the corresponding barcode produced by persistence, where the abscissa line represents the geometric scale with a logarithmic graduation, and where left (resp. right) arrows mark left- (resp. right-) infinite intervals.

— *Pictures used in the left part of the figure are courtesy of the Computer Vision Laboratory at Columbia University.*

cluster. Although weaker than the full hierarchical representation, the barcode is still informative enough to allow for an interpretation. In the present example, the 11 bars with multiplicity 1 come from the 11 B's merging into a single A around scale $2^4 = 16$. Before that, the 15 bars with multiplicity 11 come from each letter B having 16 points that get merged into a single cluster around scale $2^3 = 8$. It takes a bit of time to get used to this kind of representation, in which the actual hierarchy (who is merged with whom) is lost. Nevertheless, this is the price to pay for more stability and generality.

Persistence is indeed able to produce such barcodes for higher-dimensional topological features as well. For instance, the bottom half of Figure 0.3 shows a barcode encoding the lifespans of 'holes' across scales in the data set of Figure 0.1. To understand what is meant by this, imagine each data point being replaced by a ball of radius r at scale r. Persistence detects the holes in the resulting union of balls at every scale, and tracks their persistence across scales. Each bar in the resulting barcode corresponds to a particular hole, and it encodes its lifespan in the growing family of balls. The same can be done for voids in higher dimensions. In the example of Figure 0.3, the 2 bars with multiplicity 11 appearing at lower scales come from each letter B having 2 holes, while the long bar with multiplicity 1 appearing at larger scales comes from the letter A having a single hole. The rest of the bars indicate holes created at intermediate steps in the ball growing process, for instance in places where B's are arranged into a triangle.

Being able to detect the presence of topological features of arbitrary dimensions in data, and to represent these features as a barcode whatever their dimension, is what makes persistence an interesting tool for data visualization and analysis, and a nice complement to more classical techniques such as clustering or dimensionality reduction [183]. Besides, being able to do so in high dimensions and in a robust way, as illustrated in Figure 0.4, is an asset for applications. It is also an algorithmic challenge, as dealing with high-dimensional data requires to develop a computing machinery that scales up reasonably with the ambient dimension.

Persistence in a nutshell. The theory works at two different levels: topological, and algebraic. At the topological level, it takes as input a sequence of nested topological spaces, called a *filtration*:

$$X_1 \subseteq X_2 \subseteq \cdots \subseteq X_n. \tag{0.1}$$

Such sequences come typically from taking excursion sets (sublevel sets or superlevel sets) of real-valued functions. For instance, in the example of Figure 0.3, the filtration is composed of the sublevel sets of the distance to the point cloud, the r-sublevel set being the same as the union of balls of same radius r about the data points, for every $r \geq 0$. Here already comes a difficulty: in (0.1) we are using a finite sequence, whereas the sublevel sets of a function form a continuous 1-parameter family. While algorithms only work with finite sequences for obvious reasons, the theory is stated for general 1-parameter families. The connection between discrete and continuous families is not obvious in general, and determining the precise conditions to be put on a continuous family so that it behaves 'like' a discrete family has been the subject of much investigation, as will be reflected in the following chapters.

Given a sequence like (0.1), we want not only to compute the topological structure of each space X_i separately, but also to understand how topological features persist across the family. The right tool to do this is homology over a field, which turns (0.1) into a sequence of vector spaces (the homology groups $\mathsf{H}_*(X_i)$) connected by linear maps (induced by the inclusions $X_i \hookrightarrow X_{i+1}$):

$$\mathsf{H}_*(X_1) \longrightarrow \mathsf{H}_*(X_2) \longrightarrow \cdots \longrightarrow \mathsf{H}_*(X_n). \tag{0.2}$$

Such a sequence is called a *persistence module.* Thus we move from the topological level to the algebraic level, where our initial problem becomes the one of finding bases for the vector spaces $\mathsf{H}_*(X_i)$ that are 'compatible' with the maps in (0.2). Roughly speaking, being compatible means that for any indices i, j with $1 \leq i \leq j \leq n$, the composition

$$\mathsf{H}_*(X_i) \longrightarrow \mathsf{H}_*(X_{i+1}) \longrightarrow \cdots \longrightarrow \mathsf{H}_*(X_{j-1}) \longrightarrow \mathsf{H}_*(X_j)$$

has a (rectangular) diagonal matrix in the bases of $\mathsf{H}_*(X_i)$ and $\mathsf{H}_*(X_j)$. Then, every basis element can be tracked across the sequence (0.2), and its *birth time* b and *death time* d defined respectively as the first and last indices at which it is part of the current basis. At the topological level, this basis element corresponds to some feature (connected component, hole, void, etc.) appearing in X_b and disappearing in X_{d+1}. Its lifespan is encoded as an interval $[b, d]$ in the persistence barcode[3].

The very existence of compatible bases is known from basic linear algebra when $n \leq 2$ and from the structure theorem for finitely generated modules over a principal ideal domain when n is arbitrary (but finite) and the vector spaces $\mathsf{H}_*(X_i)$ have finite dimensions. Beyond these simple cases, e.g. when the index set is infinite or when the spaces are infinite-dimensional, the existence of compatible bases is not always assured, and when it is, this is thanks to powerful decomposition theorems from quiver representation theory. Indeed, in its algebraic formulation, persistence

[3] Some authors rather use the equivalent notation $[b, d+1)$ for the interval. We will come back to this in Chapter 1.

is closely tied to quiver theory. Their relationship will be stressed in the following chapters, but for now let us say that *quiver* is just another name for (multi-)graph, and that a *representation* is a realization of a quiver as a diagram of vector spaces and linear maps. Thus, (0.2) is a representation of the quiver

$$\underset{1}{\bullet} \longrightarrow \underset{2}{\bullet} \longrightarrow \cdots \longrightarrow \underset{n}{\bullet}$$

Computing a compatible basis is possible when the filtration is *simplicial*, that is, when it is a finite sequence of nested simplicial complexes. It turns out that computing the barcode in this special case is hardly more complicated than computing the homology of the last complex in the sequence, as the standard matrix reduction algorithm for computing homology can be adapted to work with the filtration order. Once again we are back to the question of relating the barcodes of finite (simplicial) filtrations to the ones of more general filtrations. This can be done via the stability properties of these objects.

Stability. The stability of persistence barcodes is stated for an alternate representation called *persistence diagrams.* In this representation, each interval[4] b, d is viewed as a point (b, d) in the plane, so a barcode becomes a planar multiset. The persistence of a topological feature, as measured by the length $(d - b)$ of the corresponding barcode interval b, d, is now measured by the vertical distance of the corresponding diagram point (b, d) to the diagonal $y = x$. For instance, Figure 0.5 shows the persistence diagrams associated to the filtrations of (the sublevel-sets of) two functions $\mathbb{R} \to \mathbb{R}$: a smooth function f, and a piecewise linear approximation f'. As can be seen, the proximity between f and f' implies the proximity between their diagrams $\mathsf{dgm}(f)$ and $\mathsf{dgm}(f')$. This empirical observation is formalized in the following inequality, where $\|\cdot\|_\infty$ denotes the supremum norm and $\mathrm{d_b}$ denotes the so-called *bottleneck distance* between diagrams:

$$\mathrm{d_b}(\mathsf{dgm}(f),\ \mathsf{dgm}(f')) \leq \|f - f'\|_\infty. \tag{0.3}$$

Roughly speaking, the bottleneck distance provides a one-to-one matching between the diagram points corresponding to highly persistent topological features of f and f', the topological features with low persistence being regarded as noise and their corresponding diagram points being matched to the nearby diagonal.

Stability, as stated in (0.3) and illustrated in Figure 0.5, is an important property of persistence diagrams for applications, since it guarantees the consistency of the computed results. For instance, it ensures that the persistence diagram of an unknown function can be faithfully approximated from the one of a known approximation. Or, that reliable information about the topology of an unknown geometric object can be retrieved from a noisy sampling under some reasonable noise model.

The proof of the stability result works at the algebraic level directly. For this it introduces a measure of proximity between persistence modules, called the *interleaving distance*, which derives naturally from the proximity between the functions the modules originate from (when such functions exist). In this metric, the stability result becomes in fact an isometry theorem, so that comparing persistence modules is basically the same as comparing their diagrams. From there on, persistence diagrams can be used as signatures for all kinds of objects from which persistence modules are derived, including functions but not only.

[4] We are omitting the brackets to indicate that the interval can be indifferently open, closed, or half-open.

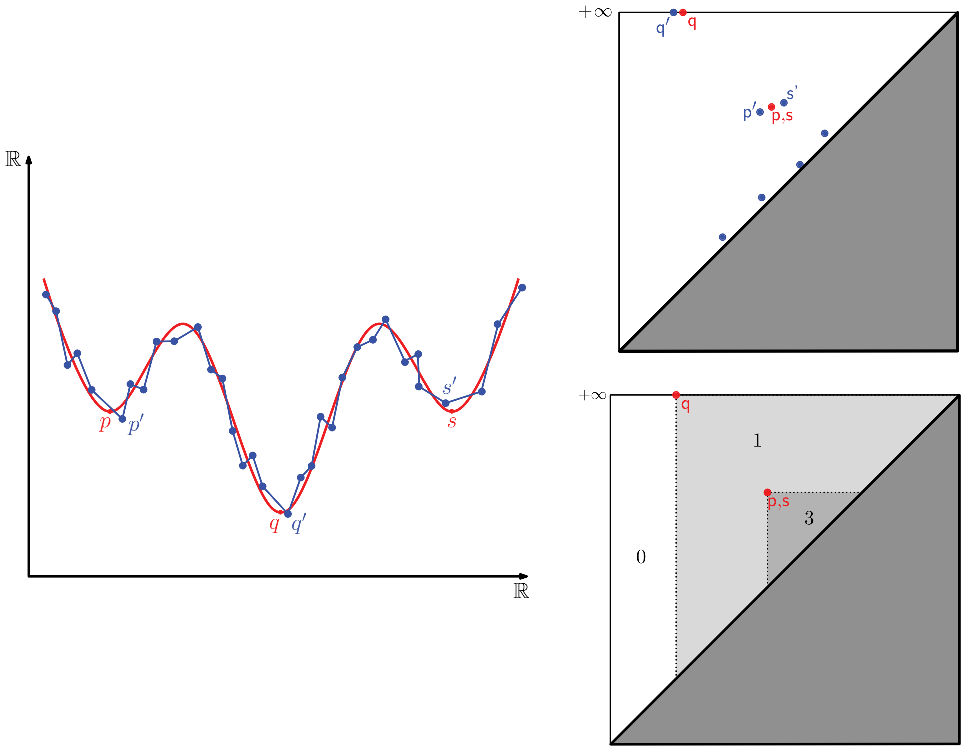

FIGURE 0.5. Left: A smooth function $f : \mathbb{R} \to \mathbb{R}$ (red) and a piecewise linear approximation f' (blue). Top-right: Superimposition of the persistence diagrams of f (red) and f' (blue). Every red diagram point (b, d) corresponds to some local minimum of f creating an independent connected component in the sublevel set of f at time b, and merging it into the component of some lower minimum at time d, as per the *elder rule*. Idem for blue points and f'. Bottom-right: The size function corresponding to the persistence diagram of f.

The isometry theorem is the cornerstone of the current theory, and its main asset for applications. It is also what makes persistence stand out of classical quiver theory.

Connections to other theories. As mentioned previously, there is a deep connection between the algebraic level of persistence and quiver theory. Meanwhile, the topological level has strong bonds with Morse theory:

- In the special case where the input filtration is given by the sublevel sets of a Morse function f, i.e. a C^∞-continuous real-valued function with non-degenerate critical points such that all the critical values are distinct, Morse theory describes when and how the topology of the sublevel sets of f changes in the filtration [195, theorems 3.1 and 3.2], thus providing a complete characterization of its persistence diagram. Persistence generalizes this analysis beyond the setting of Morse theory, to cases where the function f may not be differentiable nor even continuous, and where its domain may not be a smooth manifold nor a manifold at all.

- In a similar way, persistence for simplicial filtrations is related to the discrete version of Morse theory [130]. There are indeed filtered counterparts to the discrete gradient fields and to their associated discrete Morse complexes. These are defined on simplicial filtrations rather than on single simplicial complexes, with the expected property that the persistent homology of the filtered Morse complexes is the same as the one of the filtrations they come from. This connection has been exploited in various ways, for instance to speed up the persistence computation [198].
- Finally, the 0-dimensional aspects of persistence are related to Morse theory in a particular way [55, 213]. Given a Morse function f, the hierarchy on the local minima of f produced by persistence from the family of its sublevel sets is equivalent to the *join tree* of f. Similarly, the hierarchy on the local maxima of f produced from its superlevel sets is equivalent to the *split tree* of f. Once merged together, these two trees form the *contour tree* of f, which is the loop-free version of the *Reeb graph* and is equal to it when the domain of f is both connected and simply connected. There are also some relations between the 1-dimensional persistence of f and the loops of its Reeb graph [86, 108].

As we saw earlier, the connection between the topological and the algebraic levels of persistence happens through the use of homology, which turns sequences of topological spaces into sequences of vector spaces. Using the metaphor of a space changing over time to describe each sequence, we can view persistence as a generalization of classical homology theory to the study of time-evolving spaces. In this metaphor, persistence modules such as (0.2) are the time-dependent analogues of the homology groups, and their barcodes are the time-dependent analogues of the Betti numbers. Although somewhat restrictive, this view of the theory is convenient for interpretation.

Persistence is also a generalization of *size theory* [97, 131], whose concern is with the quantity

$$\text{rank } \mathsf{H}_0(X_i) \to \mathsf{H}_0(X_j)$$

defined for all pairs (i, j) such that $1 \leq i \leq j \leq n$, and called the *size function* of the filtration (0.1). The value of the size function at (i, j) measures the number of connected components of X_i that are still disconnected in X_j. The level sets of this function look like staircases in the plane, whose upper-left corners are the points recorded in the 0-dimensional part of the persistence diagram—see Figure 0.5 for an illustration. The stability result (0.3) appeared in size theory prior to the development of persistence, however in a form restricted to 0-dimensional homology.

Finally, the algorithmic aspects of persistence have close connections to *spectral sequences* [81]. Roughly speaking, the spectral sequence algorithm outputs the same barcode as the matrix reduction algorithm, albeit in a different order.

These connections at multiple levels bear witness to the richness of persistence as a theory.

Applications. This richness is also reflected in the diversity of the applications, whose list has been ever growing since the early developments of the theory. The following excerpt[5] illustrates the variety of the topics addressed:

- analysis of random, modular and non-modular scale-free networks and networks with exponential connectivity distribution [158],
- analysis of social and spatial networks, including neurons, genes, online messages, air passengers, Twitter, face-to-face contact, co-authorship [210],
- coverage and hole detection in wireless sensor fields [98, 136],
- multiple hypothesis tracking on urban vehicular data [23],
- analysis of the statistics of high-contrast image patches [54],
- image segmentation [70, 209],
- 1d signal denoising [212],
- 3d shape classification [58],
- clustering of protein conformations [70],
- measurement of protein compressibility [135],
- classification of hepatic lesions [1],
- identification of breast cancer subtypes [205],
- analysis of activity patterns in the primary visual cortex [224],
- discrimination of electroencephalogram signals recorded before and during epileptic seizures [237],
- analysis of 2d cortical thickness data [82],
- statistical analysis of orthodontic data [134, 155],
- measurement of structural changes during lipid vesicle fusion [169],
- characterization of the frequency and scale of lateral gene transfer in pathogenic bacteria [125],
- pattern detection in gene expression data [105],
- study of plant root systems [115, §IX.4],
- study of the cosmic web and its filamentary structure [226, 227],
- analysis of force networks in granular matter [171],
- analysis of regimes in dynamical systems [25].

In most of these applications, the use of persistence resulted in the definition of new descriptors for the considered data, which revealed previously hidden structural information and allowed the authors to draw original conclusions.

Contents. There are three parts in the book. The first part focuses on the theoretical foundations of persistence. It gives a broad view of the theory, including its algebraic, topological, and algorithmic aspects. It is divided into three chapters and an appendix:

- Chapter 1 introduces the algebraic aspects through the lense of quiver theory. It tries to show both the heritage of quiver theory and the novelty brought in by persistence in its algebraic formulation. It is supplemented with Appendix A, which gives a formal introduction to quiver representation theory and highlights its connections to persistence. Concepts such as persistence module, zigzag module, module homomorphism, interval decomposition, persistence diagram, quiver, representation, are defined in these two chapters.

[5] Much of the list was provided by F. Chazal, F. Lecci and B. Michel, who recently took an inventory of existing applications of persistence.

- Chapter 2 introduces the topological and algorithmic aspects of persistence theory. It first reviews the topological constructions that are most commonly used in practice to derive persistence modules. It then focuses on the algorithms designed to compute persistence from filtrations: the original algorithm, described in some detail, then a high-level review of its variants and extensions. Concepts such as filtration, zigzag, pyramid, persistent (co-)homology, are defined in this chapter.
- Chapter 3 is entirely devoted to the stability of persistence diagrams, in particular to the statement and proof of the *Isometry Theorem*, which is the central piece of the theory. After introducing and motivating the measures of proximity between persistence modules and between their diagrams which are used in the statement of the theorem, it develops the main ideas behind the proof and discusses the origins and significance of the result. Concepts such as interleaving distance, bottleneck distance and matching, snapping principle, module interpolation, are defined in this chapter.

The second part of the document deals with applications of persistence. Rather than trying to address all the topics covered in the aforementioned list, in a broad and shallow survey, it narrows the focus down to a few selected problems and analyzes in depth the contribution of persistence to the state of the art. Some of these problems have had a lasting influence on the development of the theory. The exposition is divided into four chapters:

- Chapters 4 and 5 introduce the problem of inferring the topology of a geometric object from a finite point sample, which was and continues to be one of the main motivations for the development of the theory. The general approach to the problem is presented in Chapter 4, along with some theoretical guarantees on the quality of the output. Algorithmic aspects are addressed in Chapter 5, which introduces recent techniques to optimize the running time and memory usage, improve the signal-to-noise ratio in the output, and handle a larger variety of input data.
- Chapter 6 focuses more specifically on the 0-dimensional version of topological inference, also known as clustering. It formalizes the connection between persistence and hierarchical clustering, which we saw earlier. It also draws a connection to mode seeking and demonstrates how persistence can be used to stabilize previously unstable hill-climbing methods. Finally, it addresses the question of inferring higher-dimensional structure, to learn about the composition of each individual cluster as well as about their interconnectivity in the ambient space. This part comes with comparatively little effort once the persistence framework has been set up.
- Chapter 7 shifts the focus somewhat and addresses the problem of comparing datasets against one another. After setting up the theoretical framework, in which datasets and their underlying structures are treated as metric spaces, it shows how persistence can be used to define descriptors that are provably stable under very general hypotheses. It also adresses the question of computing these descriptors (or reliable approximations) efficiently. Down the road, this chapter provides material for comparing shapes, images, or more general data sets, with guarantees.

The third part of the document is more prospective and is divided into two short chapters: one is on current trends in topological data analysis (Chapter 8), the other is on further developments of the theory (Chapter 9). This part gathers the many open questions raised within the previous parts, along with some additional comments and references.