An implementation of topological data analysis in clustering and classifying time series data

By Hua-Wei Huang, 2020

Disclaimer

Some contents in this file is based on my MSc dissertation; however, due to confidential reason, I only include a general illustration of the implementation of topological data analysis (TDA) to time series data. This file only contains theoretical background about TDA and is in support to the code (here). Again, this file and the codes are only to illustrate the use of TDA. Hence, the result is not related to the conclusion of my dissertation.

Table of Contents

1. Introduction

The premise of the report is to present the implementation of topological data analysis (TDA) to the ubiquitous time-series data and the fundamental background theories. Specifically, we focus on univariate time series data obtained from UCR Time Series Classification Archive and examine the performance of clustering and classification methods with and without topological-related features.

Again, this experiment is only to illustrate the use of TDA; therefore, we focus on the selected time series data, clustering and classification method for the ease of representation. We consider agglomerative hierarchical clustering and 1NN classifier to compare the performance of topological-related features. Moreover, we consider the time series dataset, "DistalPhalanxOutlineAgeGroup", which is split into training and test dataset. The dataset contains total 539 time series sequence with equal length of 80 and fall into three classes. The training dataset contains 400 time series sequences and the test dataset contains 139 sequences.

One of the areas in clustering and classification considers similarity distance measures between time series sequences to reflect the geometrical compactness between sequences (Pereira and de Mello, 2015). Nonetheless, these similarity measures could not reflect accurately with data that are in the shape of loops or holes. Topological data analysis (TDA) is a developing and recent field that combines knowledge from topology and computational geometry to analyse the underlying shapes or features of a dataset (Chazal and Michel, 2017). The main advantage of TDA is the ability in evaluating the topological and geometrical structure from a dataset.

The following figure is an illustration in the clustering results based on Euclidean distance measures and topological-related approach. One can see that the Euclidean distance separates the data based on their compactness while the topological-related approach separates them based on the shape (circle vs dots).

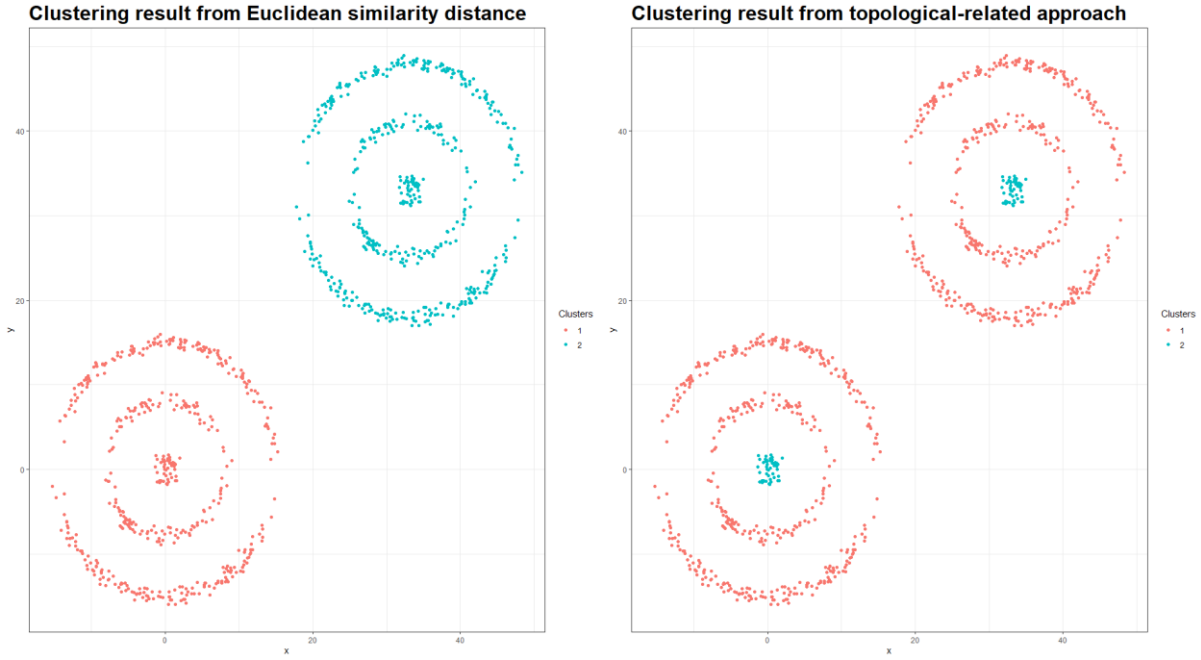

Figure 1. An illustration of clustering result from Euclidean distance approach and topological-based approach.

In this report, we assume readers have basic knowledge in similarity distance measures, clustering and classification methods. Hence, we focus on the theoretical background and process of TDA. The rest of this report is organised as follows. In Section (2), we introduce the necessary backgrounds on TDA. We also provide a brief discussion of validation index in evaluating clustering and classification results in Section (2). We describe our experiments and process of implementing TDA in Section (3). Then in Section (4), we present our experimental results and discussions. For readers interested only in the process in implementing TDA, we suggest skipping to Section (3) and also look at the Jupyter notebook which contains the codes. [link to the Jupyter notebook]

2. TDA for time series analysis

Algebraic topology is a branch of mathematics which studies topological features of high-dimensional objects and is interested in the connectivity between two spaces (Edelsbrunner and Harer, 2008; Hatcher, 2002; Rieck, 2017).

Topological data analysis (TDA) is a developing and recent field that combines knowledge from topology and computational geometry to analyse the underlying shapes or features of a dataset (Chazal and Michel, 2017). Unlike traditional data analysis that has been developed in producing valuable quantitative information, TDA is motivated in combining concepts from topology and geometry to provide remarkable qualitative information such as geometrical structure of a dataset (Pereira and de Mello, 2015; Chazal and Michel, 2017).

The theories of persistent homology are used extensively in topology data analysis in order to describe topological features of data with persistent diagram (PD). Persistent homology is established based on the foundation of algebraic topology and simplicial homology in order to study the topological features of a dataset (Rieck, 2017). In terms of a time series data, a sequence of time series is discretised into a point cloud by employing some trajectory methods such as Takens embedding. Furthermore, the point cloud is constructed into topological space to represent n-dimensional generalisation; this step employs simplicial complexes with filtration.

A simplicial complex is a representation of how data points in a topological space are connected together which is built from points, edges, or triangular faces. For instance, given a distance d in a two-dimensional space, the datapoints that are overlapped within diameter of d are connected together. When data points are connected, a trajectory is formed with shapes such as triangle or holes. As distance d increases, the number of holes and voids changes. Once a hole is formed at distance d, it is considered as a birth at d. The information extracted from simplicial complexes is presented in persistent diagram where the diagram contains the birth and death of k-dimensional holes. In this section, we illustrate the concepts behind these methods; additionally, we focus on the persistent homology based on point clouds. In general, the steps in using topological data analysis and persistent homology to cluster or classify time series data are:

Step 1 Pre-processing of the time series data

Step 2 Trajectory reconstruction (forming point clouds)

- Takens embedding theorem

Step 3 Simplicial complex and filtration

- Cech complex
- Vietoris-Rips complex

Step 4 Persistent diagram (PD)

Step 5 Features extraction from PD

Step 6 Appling clustering or classification algorithms on reconstructed data set based on step 5.

2.1. Takens theorem

Takens' theorem is a delay embedding theorem developed by Floris Takens (Takens, 1981) where the theorem provides a foundation and motivation in sliding window embedding. Takens' theorem provides a theoretical background in reconstructing dynamic attractor from time series observations. Specifically, the theory implies that the underlying dynamics of a single measured time series could be analysed by embedding the series to higher-dimensional manifold (Sauer, 2006) without losing its original topological attributes (Truong, 2017).

Before defining the Takens theorem, we highlight some basic dynamical system and embedding. Dynamical system is a mathematical rule in describing changes within a state space which is time dependent (Meiss, 2007). In terms of dynamical system, an attractor, in the simplest form, is a set of points where every state space tends to evolve towards (Milnor, 2006); additionally, a topological space is considered as an abstract state space in dynamical systems theory (Terman and Izhikevich, 2008). As a result, the manifold of dynamical system could be helpful in analysing its underlying features. A map is a representation of the evolution rule which predicts the next state from current state space value. A map from manifold $\mathcal{M}_1$ to $\mathcal{M}_2$ is noted as Φ: $\mathcal{M}_1 \longrightarrow \mathcal{M}_2$ where $\mathcal{M}_2$ is considered as an embedding of $\mathcal{M}_1$ if Φ is a diffeomorphism from $\mathcal{M}_1$ to $\mathcal{M}_2$ (Truong, 2017).

We provide the Takens theorem (Robinson, 2005; Perea, 2019) which is defined as

Theorem 1 *Let $\mathcal{M}$ be a Riemannian manifold, τ be a real number and d be an integer where $\tau > 0$ and $d \geq 2dim(\mathcal{M})$. If $\Phi \epsilon C^2(\mathbb{R} \times \mathcal{M}, \mathcal{M})$ and $F \epsilon C^2(\mathcal{M}, \mathbb{R})$ are generic, the delay map in Equation (1) is an embedding for a time series, $\varphi_p(t)$, defined by Equation (2).*

$$\begin{aligned} \varphi: \mathcal{M} &\longrightarrow \mathbb{R}^{d+1} \\ p &\mapsto (\varphi_p(0), \varphi_p(\tau), \varphi_p(2\tau), \ldots, \varphi_p(d\tau)) \end{aligned} \tag{1}$$

$$\begin{aligned} \varphi_p: \mathbb{R} &\longrightarrow \mathbb{R} \\ t &\mapsto F \circ \Phi(t, p) \end{aligned} \tag{2}$$

given that $F : \mathcal{M} \longrightarrow \mathbb{R}$ and $p \epsilon \mathcal{M}$.

Takens' theorem is extended from Whiteney's embedding theorem where Takens' theorem finds a function Φ that embeds n-dimensional manifolds into a single time-delayed signal. According to theorem 1, a n-dimensional manifold can be embedded into $\mathbb{R}^{2n+1}$. Given a time series $X = [x_1, x_2, \ldots, x_n]$, a trajectory matrix $\mathbb{X}$ which represents a point cloud with dimension d and time lag τ is

$$\mathbb{X} = \begin{bmatrix} x_{1+(d-d)\tau} & x_{1+(d-(d-1))\tau} & \cdots & x_{1+(d-2)\tau} & x_{1+(d-1)\tau} \\ x_{2+(d-d)\tau} & x_{2+(d-(d-1))\tau} & & x_{2+(d-2)\tau} & x_{2+(d-1)\tau} \\ & \vdots & \ddots & \vdots & \\ x_{n-1+(d-d)\tau} & x_{n-1+(d-(d-1))\tau} & & x_{n-1+(d-2)\tau} & x_{n-1+(d-1)\tau} \\ x_{n-(d-d)\tau} & x_{n+(d-(d-1))\tau} & \cdots & x_{n+(d-2)\tau} & x_{n+(d-1)\tau} \end{bmatrix} \tag{3}$$

2.2. Simplicial complex and filtration

In a lower dimensional space, it is easier in visualising and understanding the shape formed by the data points. Nonetheless, in a high-dimensional space, it is usually difficult in visualising and representing its topological features. As a result, simplicial complex serves as a representation of topological space in algebraic topology. One critical step in computing persistent homology is the construction of simplicial complex. Simplicial complex is a structural representation that describes the geometrics, such as, point, edges, lines, or triangles from connected components of point clouds in a topological space (Rieck, 2017; Chen 2019). Simplicial complex provides a balance invariant in analysing underlying topological features in a topological space. In this subsection, we briefly provide background associated with simplicial complex. Then, we illustrate filtration and two common simplicial complexes, Cech complex and Vietoris-Rips complex.

2.2.1. Simplices

In the context of simplicial complex, simplices are the generalisation of the components that constructs complexes. These simplices are the foundation in creating simplicial complexes (Boyd et al., 2004; Edelsbrunner, 2006b). For instance, 0-simplex represents an object with one vertex in a 0-dimensional space. 1-simplex represents an object with two vertices in 1-dimensional space. Specifically, 0-simplex is considered as a vertex and 1-simplex is an edge; this is illustrated in Figure 2. We define a k-simplex as

Definition 1 *The k-simplex is defined by $\boldsymbol{x_0}, \ldots, \boldsymbol{x_i}$ point from a set of convex combination, $\boldsymbol{\lambda_0 x_0}, \ldots, \boldsymbol{\lambda_i x_i}$ where $\sum_{n=0}^{i} \boldsymbol{\lambda_n} = \mathbf{1}$ and $\boldsymbol{k} = \boldsymbol{i}$ points.*

The concept of simplices can be applied to a higher-dimensional space. A face of a k-simplex could be considered as a subset of the vertices from k-simplex, for example, in the right plot from Figure 2, $X1 - X2$ is a 1-face (edge) in a 2-simplex. We refer readers to (Edelsbrunner, 2006b; Rieck, 2017) for more detailed description on the formation of k-simplex.

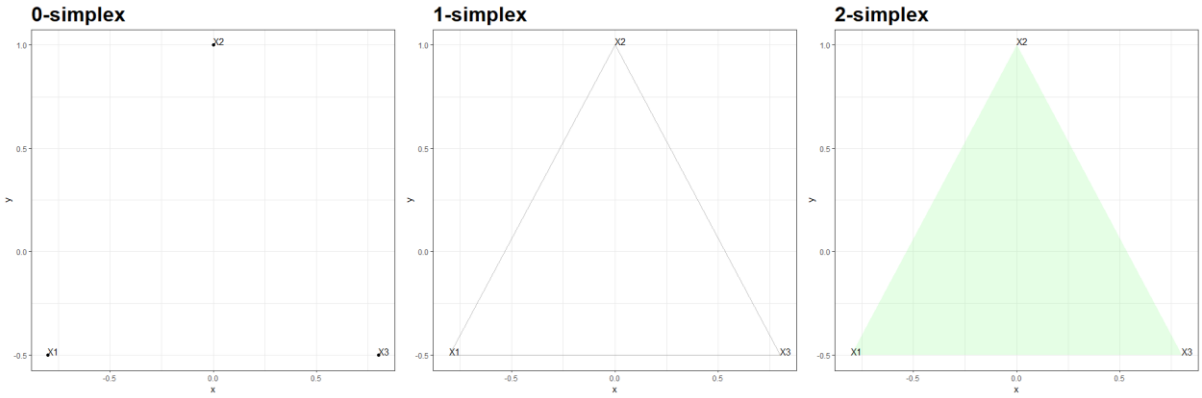

Figure 2. An Illustration of simplices formed from three points, the left is 0-simplex (points), the middle is 1-simplex (edges) and the right is 2-simplex (faces).

2.2.2. Nerve theorem

Before constructing a simplicial complex, we require a mathematical rigour in converting point clouds into computable representation in topology. The nerve theorem provides a base in this purpose through the concept of covering (Rieck, 2017). A cover is defined as

Definition 2

$$\mathcal{U} := \{U_i | i \in I\} \textit{ is a cover of } \mathbb{X} \textit{ if } \mathbb{X} \subseteq \bigcup_{i \in I} U_i \tag{4}$$

where $\mathbb{X}$ *is a topological space and* U_i *are subsets of* $\mathbb{X}$.

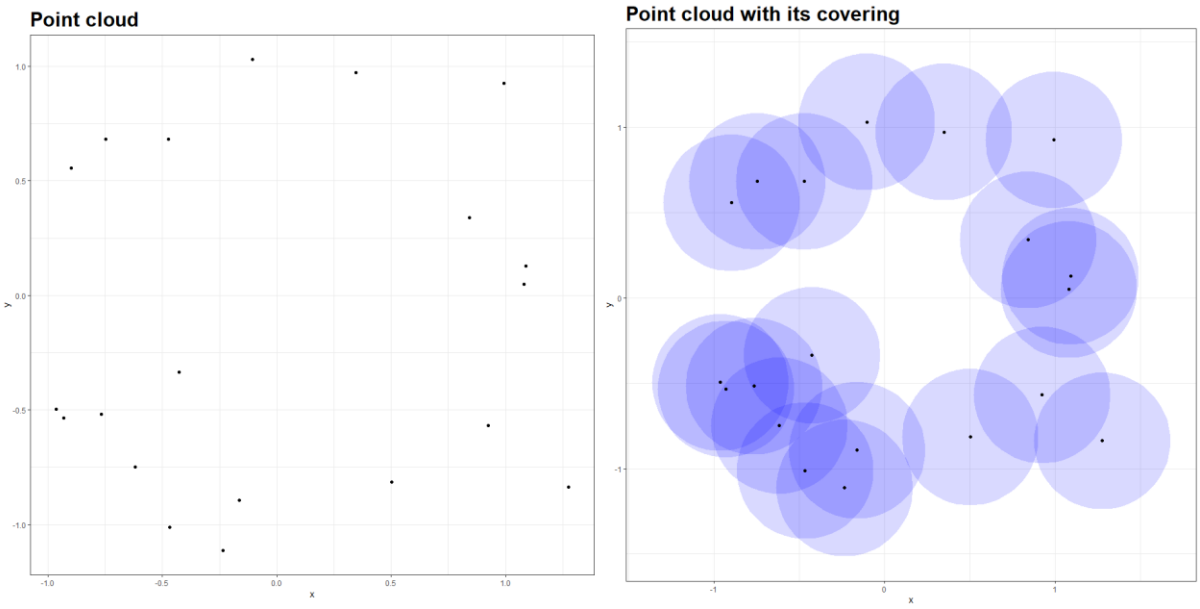

Figure 3. An illustration of point cloud in topological space and its covering. Each circle/ball has a radius of 0.4. The left is the point cloud and the right is its corresponding covering.

Figure 3 shows an example of covering from a discrete topological space. Additionally, the nerve of a covering is defined as

$$Nerve\ \mathcal{U} := \{U_i \subseteq \mathcal{U} | \bigcap U_i \neq \emptyset\} \tag{5}$$

This means that a nerve of a covering $\mathcal{U}$ includes all the non-empty subsets whose intersection is also non-empty. Based on the properties of a covering and a nerve, this leads to Nerve theorem (Chazal and Michel, 2017; Wang, 2018) which is defined as

Theorem 2 (Nerve Theorem) Let $\mathcal{U}$ be a finite family of closed, convex sets in Euclidean space. Then the nerve of $\mathcal{U}$ and the union of the sets in $\mathcal{U}$ are homotopy equivalent.

The importance of nerve theorem is that it provides a compact and global description of the relationship between groups of points that share the common properties in a topological space through covering. The theorem provides a guarantee in constructing simplicial complex in a meaningful way. Simplicial complex is obtained through nerves from covering that share the same common properties/intersections. The nerve theorem leads us to the construction of Cech complex.

2.2.3. Cech complex and Vietoris-Rips complex

The Cech complex is a type of simplicial complex constructed from a point cloud in representing the topological features of a point cloud. Ostensibly, given a set of point cloud $\mathbb{X}$ in a metric space, the construction of Cech complex follows the following steps

Step 1 Create a ball with a user-selected radius ε around each point in $\mathbb{X}$, where $\varepsilon > 0$

Step 2 $\mathbb{C}_e$ is considered as a Cech complex if $\mathbb{C}_e$ is contained in all of the balls in $\mathbb{X}$

In terms of nerve theorem and covering, the Cech complex is the nerve of all the balls centred at points in $\mathbb{X}$. Explicitly, the Cech complex is defined as (Rieck, 2017; De Silva and Carlsson, 2004)

Definition 3 *Given a point cloud* $\mathbb{X} = [\boldsymbol{x_0}, \ldots, \boldsymbol{x_m}]$ *in* $\boldsymbol{R^n}$*, and balls with radius* $\boldsymbol{\varepsilon > 0}$*, denoted as* $\boldsymbol{B_x(\varepsilon)}$*.* $\mathbb{C}_{\boldsymbol{e}}$ *is considered as a Cech complex iff* $\mathbb{C}_{\boldsymbol{e}}$ *is contained in balls where there is no empty common intersection.*

Given a set of point cloud as in Figure 4, we draw balls with fixed radius around these points and depict the simplicial complex of each simplices as points, lines, and triangles.

The Cech complex provides a reasonable tool as in capturing topological information; additionally, increasing in radius would result in a nested complex which is helpful in constructing persistence diagram (Rieck, 2017). We will illustrate this concept later. Nonetheless, the difficulty and computation in constructing Cech complex makes it infeasible in high-dimensional space. As a result, Vietoris-Rips complex is developed which is similar to Cech complex but is more computationally tractable.

Vietoris-Rips complex

Instead of checking all subsets for non-empty common intersection, Vietoris-Rips complex only considers simplices that contains all subsets whose radius is at most ε. This means that simplices are created when all the balls have pairwise intersections. The right plot from Figure 4 is an illustration of Vietoris-Rips complex. The balls in the right plot from Figure 4 do not form any common non-empty intersection; however, there are three pairwise intersections. As a result, the example in the right plot from Figure 4 is considered as a 2-simplex while it means nothing in Cech complex. The definition of Vietoris-Rips complex is

Definition 4 *Given a point cloud* $\mathbb{X} = [\boldsymbol{x_0}, \ldots, \boldsymbol{x_m}]$ *in* $\boldsymbol{R^n}$*, and balls with radius* $\boldsymbol{\varepsilon > 0}$*, denoted as* $\boldsymbol{B_x(\varepsilon)}$*.* $\boldsymbol{\mathcal{VR}}$ *is considered as a Vietoris-Rips complex iff* $\boldsymbol{\mathcal{VR}}$ *is contained in all subsets whose radius is at most* $\boldsymbol{\varepsilon}$*.*

The simplicity of Vietoris-Rips complex makes it popular in computational topology. Generally, for a given radius, ε , the Vietoris-Rips complex and Cech complex are related to each other as follows (Chazal and Oudot, 2008)

$$\mathbb{C}_e\left(\frac{1}{2}\varepsilon\right) \subseteq \mathcal{VR}(\varepsilon) \subseteq \mathbb{C}_e(\varepsilon) \tag{6}$$

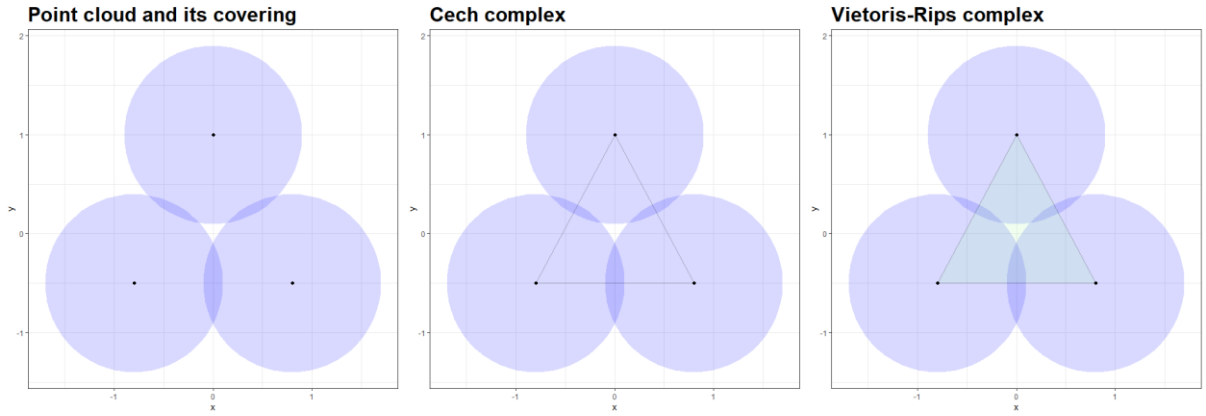

Figure 4. An illustration of point cloud, Cech complex, and Vietoris-Rips complex. The left contains point cloud and its covering, the middle includes the simplex of Cech complex which is a 1-simplex and the right is the simplex of Vietoris-Rips complex which is a 1-simplex and the right is the simplex of Vietoris-Rips complex which is a 2-simplex. These three plots are constructed under the same radius.

2.2.4. Filtration

Generally, a filtration of any space (including a simplicial complex) is an increasing nested family of subspaces (Pereira and de Mello, 2015; Chazal and Michel, 2017). Filtrations can be built directly from point clouds. For instance, Cech complexes are considered as filtrations where the topological features of the whole family of balls in Cech complexes are obtained as radius increases. Given a filtration of a simplicial complex, we can compute the persistence diagram for a point cloud. The persistent homology tracks the changes in geometrical structure of the connected components from point clouds, as shown in Figure 5, in a way that the homology is applied to detect holes in filtration. We will explain and outline concepts in persistent diagram in the next section. We define a filtration of a simplicial complex S as a nested sequence of subcomplexes (Edelsbrunner and Harer, 2008; Edelsbrunner, 2006a) as in Equation (7).

$$\emptyset = S_0 \subset S_1 \subset \ldots \subset S_n = S \tag{7}$$

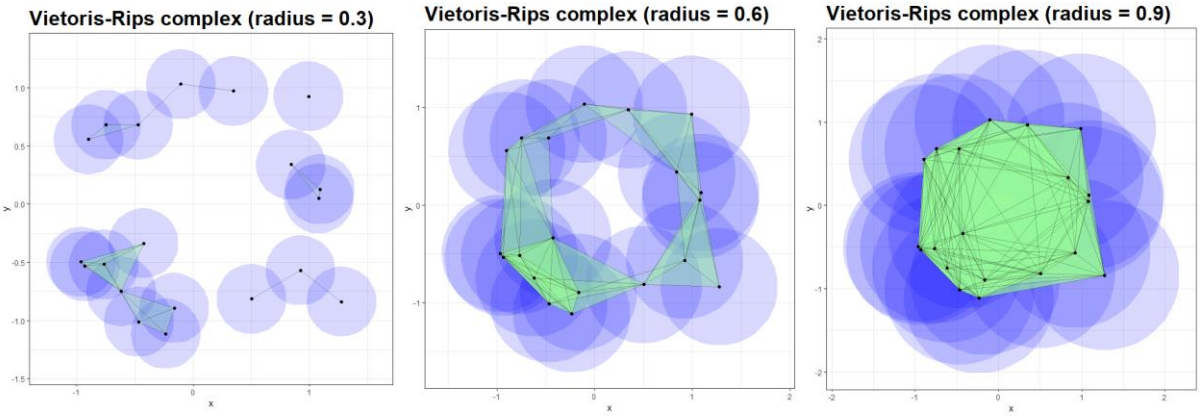

Figure 5. An illustration of filtration process as applied to a 2-dimensional point cloud under Vietoris-Rips complex as radius increases.

2.3. Persistent diagram

In the previous section, we introduced the concept of balls and two ways of constructing simplicial complex. As we described in filtration section, as the radius of the balls increase, the holes and topological features changes. A persistence diagram allows us to study how persistent these features are as the radius changes. The features that persists longer are considered to be more important (Pereira and de Mello, 2015). These changes in topological features are summarised and recorded in persistence diagram; hence, a persistence diagram is considered as a summary statistic in persistent homology. We provide a general concept behind the formation of persistence diagram in this section. For readers who are interested in explicit mathematical details, we refer the readers to (Chazal and Michel, 2017; Chazal et al., 2016).

After a filtration of a simplicial complex is constructed from a point cloud, homology detects the holes in the filtration. Each hole forms and disappears at particular radiuses. For instance, Figure 6 shows that the hole forms at radius d_1 and disappear at distance d_2. The persistence of this hole in Figure 6 is represented by point(d_1, d_2). Taking this concept further, in a more complex point cloud, we could keep track of the birth and death of the holes as the radius increases; this is represented as a persistence diagram as shown in the right plot from Figure 6.

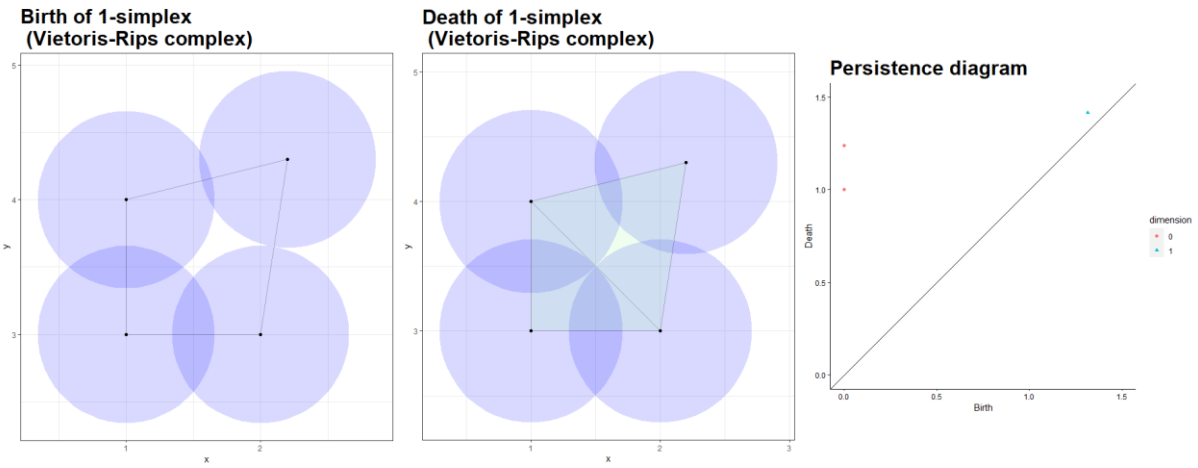

Figure 6. An illustration of the birth and death of a hole. The left is a 1-simplex at distance d_1 and the middle is the death of the hole at distance d_2. The right is the corresponding persistence diagram of the 2-dimensional point cloud. The blue dot on the persistence diagram is the birth-death coordinate of edges (1-simplex).

Generally, a persistence diagram which contains topological features is computed in the following steps (Edelsbrunner and Harer, 2010)

Step 1 Compute a simplicial complex for a point cloud

Step 2 Apply filtration to the simplicial complex

Step 3 Construct the birth and death of homology groups in step 2 as radius increases

Step 4 Output persistence diagram which contains the homology in step 3

A simple interpretation from persistence diagram in the right plot from Figure 6 is that the holes that form and die quickly are gather close to the origin (bottom left) in the persistence diagram. Additionally, the significant features are far from the diagonal. For points along the diagonal are considered as "topological noise" (Fasy et al., 2014).

2.4. Features from persistent diagram

While a persistence diagram contains topological features for a point cloud, persistent homology is equipped with distance metrics, the bottleneck distance, in measuring the similarity between two persistence diagrams. Generally, bottleneck distance measures the largest disparity between two persistence diagrams (Edelsbrunner and Harer, 2010; Rieck, 2017; Marchese, 2017). Before defining the bottleneck distance, we define L_∞-distance between two points in Equation (8) then the bottleneck distance in Equation (9).

$$\|x - y\|_\infty = \|L\|_\infty = max(|L_1|, \ldots, |L_m|) \; for \; x, y \; \epsilon \; \mathbb{R}^m. \tag{8}$$

Definition 5 *Given two persistence diagrams* $\boldsymbol{PD_1}$ *and* $\boldsymbol{PD_2}$*, the bottleneck distance between* $\boldsymbol{PD_1}$ *and* $\boldsymbol{PD_2}$ *is*

$$\boldsymbol{BT_{\infty}(PD_1, PD_2)} = \inf_{\gamma} \sup_{x \epsilon PD_1} \|x - \gamma(x)\|_{\infty} \quad (9)$$

where $\boldsymbol{\gamma}$ denotes a bijection between point sets of $\boldsymbol{PD_1}$ and $\boldsymbol{PD_2}$ and $\|\cdot\|_{\infty}$ refers to $\boldsymbol{L_{\infty}}$-distance which measures the maximum distance between two points over all dimensions.

The bottleneck distance is based on a bijection between points in each persistence diagram. The advantage of bottleneck distance is that it is insensitive to the number of points between the two persistence diagrams (Rieck, 2017).

Persistence image

In order to perform statistical analysis and machine learning from persistence diagram, one recent approach proposed by Adams et al. (2017) is to convert information from persistence diagram into a finite dimensional vector (Lacombe et al., 2018). Persistence image is introduced by Adams et al. (2017) which is a method in vectorising the information from persistence diagram. The formation of a persistence image is summarised in the following steps

Step 1 Transform multiset in birth-death coordinates from a persistent diagram, we denote as $T(\text{BD})$ where BD is a birth-death coordinates

Step 2 Map the transformed coordinates to an integrable function named persistence surface

- The persistence surface is the weighted sum of Gaussian functions as defined in Definition (6)

Step 3 Discretise relevant subdomain of persistence surface, ρ_{BD}

Step 4 Reduce persistence surface from step 3 to a finite-dimensional vector by integrating ρ_{BD} over each region in the discretisation

We provide a general definition of persistence image in Definition (7) and refer readers to (Adams et al., 2017) for details. Let BD be a birth-death coordinates in a PD which indicates the birth and death of a simplex. $T(\text{BD})$ is the transformed multiset in birth-death coordinates from a persistence diagram. Additionally, $\emptyset_u(z)$ is a function of differentiable probability distribution where normalised symmetric Gaussian is used with mean u and variance σ^2. A persistence surface is defined as

Definition 6 *For a BD in a PD, the corresponding persistence surface $\boldsymbol{\rho_{BD}}: \mathbb{R}^2 \longrightarrow \mathbb{R}$ is described as the function*

$$\boldsymbol{\rho_{BD}(z)} = \sum_{u \epsilon T(BD)} \boldsymbol{f(u)\emptyset_u(z)} \quad (10)$$

where $\boldsymbol{f}(\cdot)$ is a weighting function.

Definition 7 *Given a BD from a $\boldsymbol{PD}$, the persistence image of $\boldsymbol{BD}$ is the collection of pixels $\boldsymbol{I(\rho_{BD})_p} = \iint_p \boldsymbol{\rho_{BD} dydx}$*

According to Adams et al., (2017), the persistence image is created by assigning each boxes/pixels the corresponding integral of ρ_{BD} over that area. The discretised area/region from step 3 contains a fixed number of n boxes/pixels which forms a grid.

2.5. Validation index

Clustering

Liu et al. (2010) categorised clustering validation into two classes which are internal clustering validation and external clustering validation. Generally, the internal validation criteria evaluate the quality of clustering without any external information while external validation criteria have the knowledge of actual classes of a dataset. In this section, we illustrate the theoretical background of the external evaluation indices, Rand index and Adjusted Rand index.

We assume that $CL = \{c_1, \ldots, c_K\}$ and $CR = \{cr_1, \ldots, cr_K\}$ are two sets of clustering result with K objects. Before defining the index, we introduce some concepts in comparing pair-wise results (Halkidi et al., 2001) which we list below

- TP_{pair} (True positive): the number of pairs of points that belong to the same label in both CL and CR
- TN_{pair} (True negative): the number of points that do not belong to the same label in CL nor in CR
- FP_{pair} (False positive): the number of points that belong to the same label in CR but not in CL
- FN_{pair} (False negative): the number of points that belong to the same label in CL but not in CR

It is important to highlight that the Rand index and Adjusted Rand index are used as an evaluation for comparing clustering. Therefore, the above definitions are based on "pair-wise" points (Manning et al., 2008) and the indices consider both class label and cluster label. Unlike the classification evaluation index, accuracy, Rand index is not sensitive to cluster naming. We outline a matrix for illustrating Rand index in Table (1) and defined Rand index in Equation (11).

Table 1. An illustration of the values considered for Rand index

TP_{pair}: same class + same cluster	FN_{pair}: same class + different cluster
FP_{pair}: different class + same cluster	TN_{pair}: different class + different cluster

Rand Index

$$RI = \frac{TP_{pair} + TN_{pair}}{TP_{pair} + FP_{pair} + FN_{pair} + TN_{pair}} \tag{11}$$

Adjusted Rand Index

The Adjusted Rand index is the corrected version of the Rand index which considers the Permutation model for clustering. The permutation model generates an expected similarity for all pair-wise comparisons between clustering. An Adjusted Rand index that is closer to 1 suggests a better clustering result.

Classification

With the similar definition listed in Section for Clustering but not pair-wise points and the indices do not consider clusters, we defined accuracy (Olson and Delen, 2008) as

Accuracy

$$\frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

3. Methodology

The main purpose of our experiment is to demonstrate the use of TDA in clustering and classifying time series data. Therefore, with the background from Section (2), we proposed pipelines/workflows in clustering and classifying time series data with topological features. Our proposed pipelines are outlined in Figure 7.

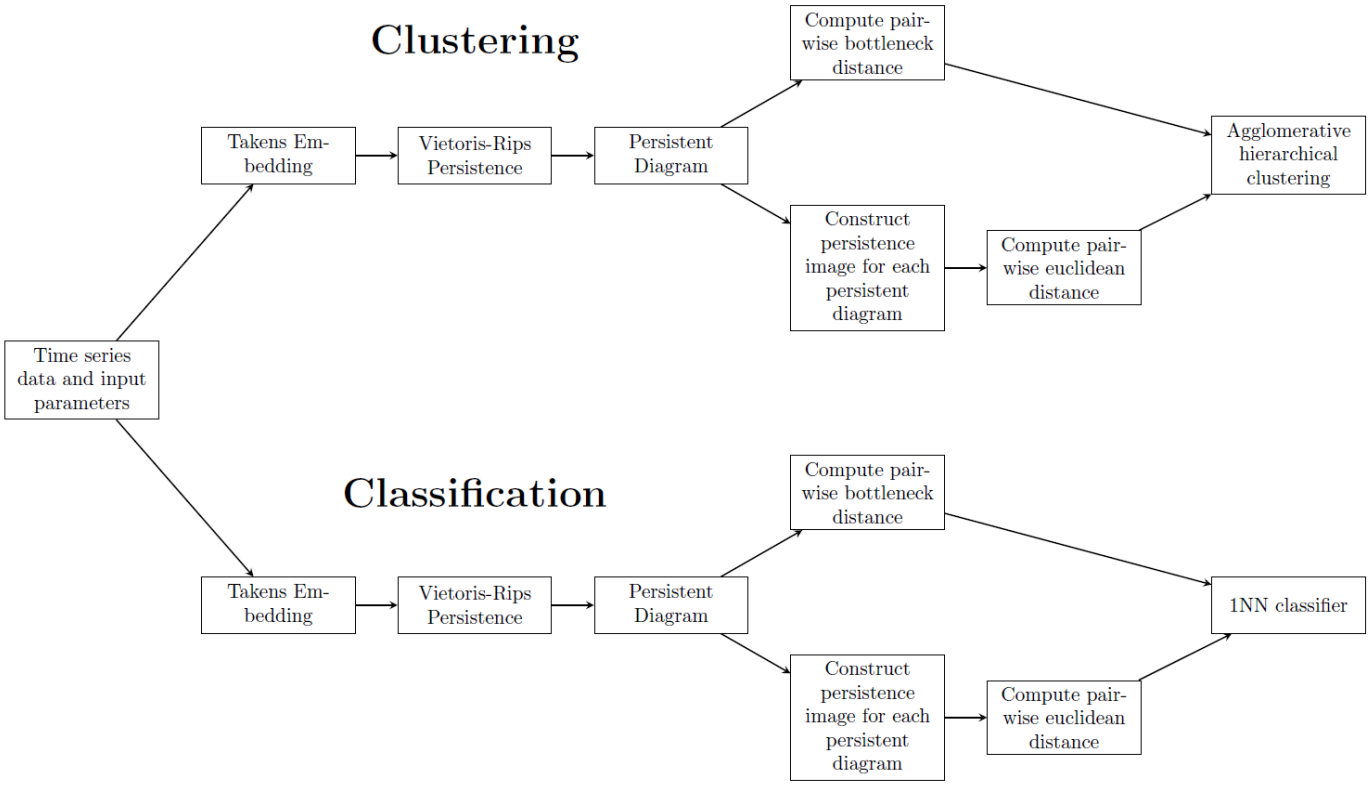

Figure 7. Pipelines of the proposed clustering and classification approaches

As mentioned in the introduction, we consider agglomerative hierarchical clustering and 1NN classifier in this experiment. Both methods depend on the notion of similarity distance measure, as a result, we compute the bottleneck distance matrix from persistent diagrams. In addition, we represent our persistent diagram in the form of persistence image and consider the persistence image as a new representation of each raw time series sequence. We compute the Euclidean distance matrix of the persistence images and pass this to clustering and classification.

In order to compare the performance of TDA in clustering and classification, we also compute Euclidean distance matrix from original raw time series data. We evaluate clustering results of these three distance matrices with Adjusted Rand index and classification results with accuracy. These results are outlined in the next section.

4. Results and discussion

In this section, we present the validation indices for clustering and classification results as proposed in Section (3); in addition, we utilised UMAP, a dimension reduction method, in representing the clustering results in two-dimension.

Table 2. Validation indices for clustering and classification results

	Euclidean distance (original data)	Euclidean distance (persistence image)	Bottleneck distance
Adjusted Rand Index	0.0310	0.2392	0.2280
Accuracy	0.6259	0.6403	0.6331

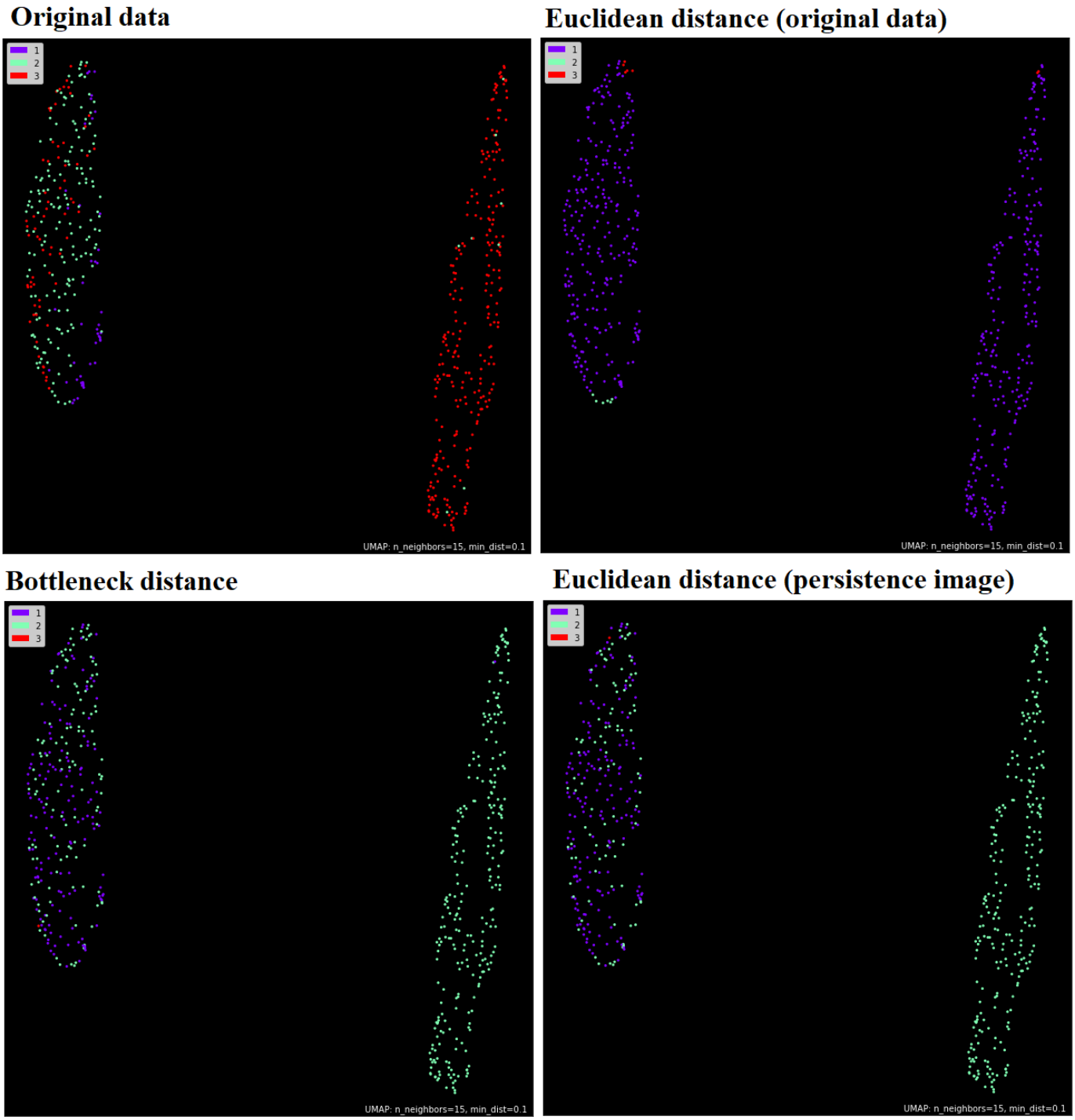

Figure 8. Visualisation of time series data based on UMAP for different clustering results

According to Table (2), one can see that the approaches related to topological features have better performances comparing to traditional approach with Euclidean distance. Nonetheless, even the topological features have a better performance, the adjusted rand index is around 0.2 which represents the clustering results do not accurately reflect the actual clustering labels. To illustrate this point, we visualise the clustering results in Figure 8 with UMAP.

Figure 8 shows original time series data in two-dimension and it is coloured based on clustering labels of different approach. From the top-left plot (original data), datapoints of three different cluster labels are located close to each other; therefore, the challenging part in clustering this particular time series dataset lies in separating these datapoints. The topological-related approaches have some ability in separating these cluster labels as they consider different aspects of the datasets comparing to traditional Euclidean distance; nonetheless, even the topological-related approaches have a better performance, they fail in clustering one of the cluster label.

The design of this experiment is only to provide readers with some insights into the implementation of TDA to time series data; therefore, we do not conclude or suggest whether TDA could consistently provide a better performance comparing to traditional approach. Nonetheless, in some particular cases, TDA does have its advantages due to the consideration of geometrical shapes.

Reference

Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F. and Ziegelmeier, L., 2017. Persistence images: A stable vector representation of persistent homology. *The Journal of Machine Learning Research*, *18*(1), pp.218-252.

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

Chazal, F. and Michel, B., 2017. An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists. *arXiv preprint arXiv:1710.04019*.

Chazal, F. and Oudot, S.Y., 2008, June. Towards persistence-based reconstruction in Euclidean spaces. In *Proceedings of the twenty-fourth annual symposium on Computational geometry* (pp. 232-241).

Chazal, F., De Silva, V., Glisse, M. and Oudot, S., 2016. *The structure and stability of persistence modules*. Springer.

Chen, R., 2019. Topological Data Analysis for Clustering and Classifying Time Series.

De Silva, V. and Carlsson, G.E., 2004. Topological estimation using witness complexes. *SPBG*, *4*, pp.157-166.

Edelsbrunner, H. and Harer, J., 2008. Persistent homology-a survey. *Contemporary mathematics*, *453*, pp.257-282.

Edelsbrunner, H. and Harer, J., 2010. *Computational topology: an introduction*. American Mathematical Soc..

Edelsbrunner, H., 2006a. *Persistent Homology*.

Edelsbrunner, H., 2006b. *Simplicial Complexes*.

Fasy, B.T., Lecci, F., Rinaldo, A., Wasserman, L., Balakrishnan, S. and Singh, A., 2014. Confidence sets for persistence diagrams. *The Annals of Statistics*, *42*(6), pp.2301-2339.

Halkidi, M., Batistakis, Y. and Vazirgiannis, M., 2001. On clustering validation techniques. *Journal of intelligent information systems*, *17*(2-3), pp.107-145.

Hatcher, A., 2002. *Algebraic topology / Allen Hatcher.*, Cambridge University Press.

Lacombe, T., Cuturi, M. and Oudot, S., 2018. Large scale computation of means and clusters for persistence diagrams using optimal transport. In *Advances in Neural Information Processing Systems* (pp. 9770-9780).

Liu, Y., Li, Z., Xiong, H., Gao, X. and Wu, J., 2010, December. Understanding of internal clustering validation measures. In *2010 IEEE International Conference on Data Mining* (pp. 911-916). IEEE.

Manning, C.D., Schütze, H. and Raghavan, P., 2008. *Introduction to information retrieval*. Cambridge university press.

Marchese, A., 2017. Data Analysis Methods using Persistence Diagrams.

Meiss, J., 2007. Dynamical systems. *Scholarpedia*, *2*(2), p.1629.

Milnor, J.W., 2006. Attractor. *Scholarpedia*, *1*(11), p.1815.

Olson, D.L. and Delen, D., 2008. *Advanced data mining techniques*. Springer Science & Business Media.

Perea, J.A., 2019. Topological time series analysis. *Notices of the American Mathematical Society*, *66*(5).

Pereira, C.M. and de Mello, R.F., 2015. Persistent homology for time series and spatial data clustering. *Expert Systems with Applications*, *42*(15-16), pp.6026-6038.

Rieck, B., 2017. *Persistent Homology in Multivariate Data Visualization* (Doctoral dissertation, Ruprecht-Karls-Universität Heidelberg).

Robinson, J.C., 2005. A topological delay embedding theorem for infinite-dimensional dynamical systems. *Nonlinearity*, *18*(5), p.2135.

Sauer, T.D., 2006. Attractor reconstruction. *Scholarpedia*, *1*(10), p.1727.

Takens, F. (1981) Detecting strange attractors in turbulence. Lecture Notes in Mathematics 898, Berlin:Springer-Verlag

Terman, D.H. and Izhikevich, E.M., 2008. State space. *Scholarpedia*, *3*(3), p.1924.

Truong, P., 2017. An exploration of topological properties of high-frequency one-dimensional financial time series data using TDA.

Wang, Y., 2018. *Chapter 2: Simplicial Complex Topics In Computational Topology: An Algorithmic View*.