VŠB – Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science

A Fuzzy Approach for Topological Data Analysis

PHD THESIS

2022 Quang-Thinh Bui

VŠB – Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Computer Science

A Fuzzy Approach for Topological Data Analysis

2022 Quang-Thinh Bui

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

VŠB – Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Computer Science

A Fuzzy Approach for Topological Data Analysis

Ph.D. Thesis; Delivered in January 2022

Doctoral Study Programme:
P1807 – Computer Science, Communication Technology and Applied Mathematics

Field of study:
1801V001 – Informatics

Ph.D. Student: **Quang-Thinh Bui**
VŠB – Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
17. listopadu 2172/15, 708 00 Ostrava, Czech Republic
quang.thinh.bui.st@vsb.cz, qthinhbui@gmail.com

Supervisors: **Prof. RNDr. Václav Snášel, CSc.**
VŠB – Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
17. listopadu 2172/15, 708 00 Ostrava, Czech Republic
vaclav.snasel@vsb.cz

Assoc. Prof. Vo Dinh Bay, PhD.
HUTECH University
Faculty of Information Technology
475A Dien Bien Phu, Binh Thanh Dist, Ho Chi Minh City, Vietnam
vd.bay@hutech.edu.vn

I hereby declare that I have independently written my doctoral thesis on the theme of "A Fuzzy Approach for Topological Data Analysis" under the guidance of the postgraduate supervisors. All technical literature and other sources of information are quoted and detailed in the list of thesis references. The results in this thesis are my original researches under the supervision of Professor Václav Snášel and Associate Professor Vo Dinh Bay at Faculty of Electrical Engineering and Computer Science, VŠB – Technical University of Ostrava. As the author of the doctoral thesis, I declare that I have not infringed any copyright regarding the creation of this thesis.

Ostrava, January 2022

First of all, I would like to express my sincerest thanks to my supervisor, Professor Václav Snášel, for always supervising, helping, encouraging me throughout my research. I greatly appreciate all his contributions and supports in all aspects during my study at VŠB – Technical University of Ostrava.

Next, I would like to express my honest thanks to my co-supervisor, Associate Professor Vo Dinh Bay, who has always accompanied, helped, and encouraged me from the first days of entering scientific research. His constant and timely supports have been the primary driving force for me to keep going in this challenging research undertaking.

Moreover, my two professors are the greatest motivation and inspiration for me to effectively complete my doctoral thesis on time with the best possible results. It is no big deal to say that this work would not happen without them.

Finally, I would like to thank my family, colleagues, and friends who always have enthusiastically supported, actively encouraged, and created the most favorable conditions for completing my thesis and finishing my doctoral course better than expected.

Abstrakt

Geometrie a topologie se stávají silnějšími a dominantnějšími v analýze dat díky svým vynikajícím vlastnostem. Nedávno se objevila jako slibná výzkumná oblast, známá jako topologická analýza dat (TDA), pro moderní informatiku. V posledních letech je algoritmus Mapper, vynikající představitel TDA, stále více doplněn o stabilizovaný teoretický základ a praktické aplikace a rozmanité, intuitivní a uživatelsky přívětivé implementace. Z teoretického hlediska je algoritmus Mapper stále fuzzy shlukovací algoritmus se schopností vizualizace extrahovat souhrn tvaru dat. Jeho výsledky jsou však stále velmi citlivé na volbu parametrů, včetně rozlišení a funkce. Proto je potřeba výrazně snížit závislost na jeho parametrech. Tato myšlenka je vzrušující a lze ji vyřešit díky vynikajícím vlastnostem fuzzy shlukování. Schopnost shlukování Mapperu je stále silnější díky podpoře známých technik. Proto se očekává, že tato kombinace užitečně a účinně vyřeší některé problémy, se kterými se setkáváme v mnoha oblastech.

Hlavním výzkumným cílem této práce je přiblížit TDA pomocí fuzzy teorie a vytvořit mezi nimi vzájemné vztahy z hlediska shlukování. Explicitně řečeno, algoritmus Mapper představuje TDA a algoritmus Fuzzy C-Means (FCM) představuje fuzzy teorii. Jsou kombinovány, aby podpořily své výhody a překonaly své nevýhody. Na jedné straně algoritmus FCM pomáhá algoritmu Mapper zjednodušit výběr parametrů pro získání nejinformativnější prezentace a je ještě efektivnější při shlukování dat. Na druhé straně je algoritmus FCM vybaven vynikajícími vlastnostmi algoritmu Mapper pro zjednodušení a vizualizaci dat pomocí kvalitativní analýzy. Tato práce se zaměřuje na dobývání a dosažení následujících cílů: (1) Shrnutí teoretických základů a praktických aplikací Mapperova algoritmu v toku literatury s vylepšenými verzemi a různými implementacemi. (2) Optimalizace volby pokrytí algoritmu Mapper ve směru automatického rozdělení rozsahu filtru do nepravidelných intervalů s náhodně se překrývajícím procentem pomocí algoritmu FCM. (3) Vytvoření nové metody pro těžbu dat, která může vykazovat stejnou schopnost shlukování jako algoritmus FCM a odhalit některé smysluplné vztahy vizualizací globálního tvaru dat poskytovaných algoritmem Mapper.

Klíčová slova: Topologická Analýza Dat, Datový Tvar, Fuzzy Shlukování, Mapper Algoritmus, Fuzzy Mapper Algoritmus, Tvar Fuzzy C-Means Algoritmus

Abstract

Geometry and topology are becoming more powerful and dominant in data analysis because of their outstanding characteristics. It has emerged recently as a promising research area, known as Topological Data Analysis (TDA), for modern computer science. In recent years, the Mapper algorithm, an outstanding TDA representative, is increasingly completed with a stabilized theoretical foundation and practical applications and diverse, intuitive, user-friendly implementations. From a theoretical perspective, the Mapper algorithm is still a fuzzy clustering algorithm, with a visualization capability to extract the shape summary of data. However, its outcomes are still very sensitive to the parameter choice, including resolution and function. Therefore, there is a need to reduce the dependence on its parameters significantly. This idea is exciting and can be solved thanks to the outstanding characteristics of fuzzy clustering. The Mapper clustering ability is getting more potent by the support from well-known techniques. Therefore, this combination is expected to usefully and powerfully solve some problems encountered in many fields.

The main research goal of this thesis is to approach TDA by fuzzy theory to create the interrelationships between them in terms of clustering. Explicitly speaking, the Mapper algorithm represents TDA, and the Fuzzy C-Means (FCM) algorithm represents fuzzy theory. They are combined to promote their advantages and overcome their disadvantages. On the one hand, the FCM algorithm helps the Mapper algorithm simplify the choice of parameters to obtain the most informative presentation and is even more efficient in data clustering. On the other hand, the FCM algorithm is equipped with the outstanding features of the Mapper algorithm in simplifying and visualizing data with qualitative analysis. This thesis focuses on conquering and achieving the following aims: (1) Summarizing the theoretical foundations and practical applications of the Mapper algorithm in the flow of literature with improved versions and various implementations. (2) Optimizing the cover choice of the Mapper algorithm in the direction of dividing the filter range automatically into irregular intervals with a random overlapping percentage by using the FCM algorithm. (3) Constructing a novel method for mining data that can exhibit the same clustering ability as the FCM algorithm and reveal some meaningful relationships by visualizing the global shape of data supplied by the Mapper algorithm.

Keywords: Topological Data Analysis, Data Shape, Fuzzy Clustering, Mapper Algorithm, Fuzzy Mapper Algorithm, Shape Fuzzy C-Means Algorithm

Contents

List of symbols and abbreviations

TDA	–	Topological Data Analysis
FCM	–	Fuzzy C-Means
FM	–	Fuzzy Mapper
KDE	–	Kernel Density Estimation
SVD	–	Singular Value Decomposition
PCA	–	Principal Component Analysis
t-SNE	–	t-Distributed Stochastic Neighbor Embedding
UMAP	–	Uniform Manifold Approximation and Projection
MM	–	Multiscale Mapper
PM	–	Parallel Mapper
MOG	–	Mapper on Graph
GNU	–	GNU's Not Unix!
BM	–	Ball Mapper
MIT	–	Massachusetts Institute of Technology
SFCM	–	Shape Fuzzy C-Means
EM	–	Ensemble Mapper
DMG	–	Deep Graph Mapper
GNN	–	Graph Neural Networks
TTM	–	Two-Tier Mapper
MIMA	–	Mapper-Induced Manifold Alignment
SSMA	–	Semi-Supervised Manifold Alignment
MBC	–	Mapper Based Classifier
AE	–	Autoencoder
DBSCAN	–	Density-Based Spatial Clustering of Applications with Noise

List of Figures

List of Tables

1 Introduction

Geometry and topology have emerged recently as a new impetus for modern computer science [1]. These frameworks are considered very natural techniques for studying massive and high-dimensional data. While geometry studies distance functions, topology examines shape invariance under any continuous transformation. The idea of combining two classical mathematical fields in data analysis comes from the point of view that point clouds are finite patterns derived from a geometric object with noise [2]. Therefore, efficient tools from different branches of geometry and topology are mobilized to study cloud datasets to discover unique properties or relationships from their topological structures. If geometry is used primarily as quantitative mathematics to focus on local networks, topology effectively provides a qualitative mathematical approach to a global data network. Geometric and topological methods allow it to extract a summarized or compressed representation for all the features of highly complex data to explore effectively and quickly specific patterns and relationships in that data [3].

Overall, the contributions of Chapter 1 can be summarized as follows:

[C1.1] Introducing TDA briefly to clarify the motivations and goals of this thesis in the context of approaching TDA by fuzzy theory.

[C1.2] Presenting concisely the fundamental theory required for TDA and fuzzy clustering to understand the expertise provided in the following chapters.

Moreover, the rest of this chapter is organized as follows. Section 1.1 introduces TDA and its outstanding representative, the Mapper algorithm. Section 1.2 deals with the motivations and goals of this thesis when approaching TDA by fuzzy theory. Section 1.3 presents the thesis structure based on its outlined goals. Section 1.4 briefly recalls the fundamental theory required for TDA and fuzzy clustering to understand the following chapters. Finally, Section 1.5 summarizes and discusses the main points presented in this chapter.

1.1 Topological Data Analysis

TDA is a low-lying interference area between algebraic topology, computational geometry, statistics, computer science, data analysis, and other related fields [4, 5]. It marks a significant turning point in the research transformation from theory to the application of geometry and topology [6, 7]. It aims to use their characteristics to construct techniques for handling and analyzing high-dimensional data. In other words, TDA can be seen as a method to gain insight into point clouds as a geometric object and extract meaningful information based on their "shape". In this area, the successful strategy stems from a straightforward but meaningful slogan "Data has shape, shape has meaning, meaning drives value" [8]. TDA focuses on data shape with local and global structures at various scales. The shape is often considered the focal point that helps analysts select appropriate methods for analyzing data. It is also emphasized

by topology as the most essential and meaningful data characteristics [9]. TDA has proven to be very effective against noisy high-dimensional data. Furthermore, although this method can be sensitive to the effects of incomplete data, it is still effective when distinguishing between datasets that have different "shapes" [10].

TDA has developed enormously in the last decade with the continuous growth of two major fundamental streams, those focusing on persistent homology [11, 12] and the Mapper algorithm [13]. While persistent homology is robust concerning small perturbations in data and provides a compact representation for studying its qualitative features with complex structure [4], the Mapper algorithm can summarize meaningful insights and valuable topological information from the high-dimensional point cloud [14]. This algorithm was first published in 2007 [13], with inspiration from spatial clustering based on the classical discrete Morse theory [15, 16], through functions defined based on all data points. However, it attracted great attention among the scientific community with the well-known work of Lum et al. published in 2013 [17], which demonstrated this approach's effectiveness and robustness in many real-life areas. Its development was motivated by the impossibility of visualizing and recognizing the structure of massive datasets, even when working with low-dimensional projections. This algorithm is designed to transform massive high-dimensional datasets into visual simplicial complexes with far fewer vertices to capture useful geometric and topological information obscured by their immense size, with various resolutions. Moreover, it is firmly constructed on the data's topological characteristics, including independence in coordinates, invariance to deformations, and compressed visualization with meaningful and helpful qualification [1]. It skillfully and efficiently exploits these characteristics to understand the topological structure and data shape.

The Mapper algorithm can be considered an advantageous method in clustering, reducing dimensionality, and making visualization possible to gain insight into the data shape. Its applications in science and life often focus on the following main aspects:

(1) Clustering data with the ability to discover exciting and meaningful structures that traditional techniques cannot detect.

(2) Selecting features that best characterize the data and efficiently explain the model.

(3) Visualizing the data shape effectively using a compressed graph representation through the customizable filter and resolutions.

The success of this method is reflected in many scientific areas, especially biomedicine [18, 19]. The Mapper algorithm's two promising areas currently being explored are the brain with neuroscience [20, 21, 22, 23] and gene structure in pathology [24, 25, 26, 27]. In addition, this algorithm is also rapidly becoming an important focus and potential framework for deep learning and artificial intelligence [28, 14, 29].

1.2 Motivation and Goal

The Mapper algorithm creates a topological approximation for a metric space via a continuous function to a low-dimensional space. The continuous function is mainly used to decompose data into overlapping sets, and a clustering algorithm is then carried out in each of them to construct clusters. After that, considering each cluster as a node, the graph is generated by connecting clusters in neighboring sets by an edge if their overlap is nonempty. This algorithm can generally return an abstract simplicial complex that is considered an easy and convenient method to visualize a topological summary of data. The original Mapper's resolution parameters include the length of small intervals and the overlapping percentage between consecutive intervals. After that, the length of intervals can be replaced by the number of intervals because of their negative correlations. When the overlapping percentage is fixed, the shorter the length of small intervals is, the larger the number of intervals is, and vice versa. The cover choice is very sensitive to the algorithm output. The cover is well-chosen if the output is the most informative, depending on the user's perspective. Covering the filter range by regular intervals and the same overlapping percentage is one weak point in the cover choice. This choice inadvertently causes unnatural phenomena in the covering method, which can be improved by using a clustering process. However, based on the overlap of clusters, there are two types of clustering, namely exclusive and overlapping. In exclusive clustering, each object belongs to exactly one cluster. In contrast, each object can belong to two or more clusters in overlapping clustering. Because of the flexibility in clustering and the compatibility in overlapping, the overlapping approach is a promising bright candidate to help the Mapper algorithm obtain a well-chosen cover more naturally. With this improvement, the filter range is covered by intervals that are not necessarily regular and the overlapping percentage that is not necessarily equal.

From a theoretical perspective, the Mapper algorithm is still a fuzzy clustering algorithm, with a visualization capability to extract the shape summary of data. However, its results are still very sensitive to the choice of parameters such as resolution and function. Therefore, there exists a need to reduce the dependence on parameters for this algorithm significantly. This problem is interesting and can be solved thanks to the outstanding characteristics of fuzzy clustering. On the one hand, the clustering ability of the Mapper algorithm is getting more powerful by the support from well-known techniques of fuzzy clustering. On the other hand, fuzzy clustering is enhanced by outstanding features in simplifying and visualizing data with qualitative analysis. This combination is expected to usefully and powerfully solve some problems encountered in many fields, especially in bioinformatics and neuroscience.

In general, the main research goal of the thesis is to approach TDA by fuzzy theory to create the interrelationships between them in terms of clustering. Explicitly speaking, the Mapper algorithm represents TDA, and the FCM algorithm represents fuzzy theory. They are combined to promote their advantages and overcome their disadvantages. On the one hand, the FCM algorithm helps the Mapper algorithm simplify the choice of parameters to obtain the most

informative presentation and is even more efficient in data clustering. On the other hand, the FCM algorithm is equipped with the Mapper algorithm's outstanding features in simplifying and visualizing data with qualitative analysis. In summary, the research goal of this study is elucidated through the following aims:

[A1] **Aim 1**: Summarizing the theoretical foundations and practical applications of the Mapper algorithm in the flow of literature with improved versions and various implementations.

[A2] **Aim 2**: Optimizing the cover choice of the Mapper algorithm in the direction of dividing the filter range automatically into irregular intervals with a random overlapping percentage by using the FCM algorithm.

[A3] **Aim 3**: Constructing a novel method for mining data that cannot only exhibit the same clustering ability as the FCM algorithm, but also reveal some meaningful relationships through visualizing the global shape of data supplied by the Mapper algorithm.

1.3 Thesis Structure

Based on the motivations and goals presented in the previous section, the thesis is structured into five chapters, specifically as follows:

[C1] Chapter 1, titled "Introduction", presents the motivations and goals to achieve the thesis. Moreover, it introduces some background on TDA and fuzzy clustering for understanding in the following chapters.

[C2] Chapter 2, titled "Mapper Algorithm", reviews the algorithm with specific descriptions and intuitive, easy-to-understand illustrations. Its variations and applications are also presented over time systematically and thoroughly. At the same time, the popular available packages that activate this algorithm as the core of their operations are also briefly introduced. Furthermore, its current limitations are also discussed to guide future research and development.

[C3] Chapter 3, titled "Fuzzy Mapper Algorithm", proposes a novel algorithm based on the foundation of the Mapper algorithm to solve the problem of automating in dividing cover intervals with an arbitrary overlapping percentage. Three real-world datasets, including Unit Circle, Reaven and Miller Diabetes, and NKI Breast Cancer, are implemented to demonstrate this algorithm's effectiveness. The experimental results are analyzed and compared with the original method, the Mapper algorithm, through the output visualization and the silhouette coefficient score in the clustering evaluation.

[C4] Chapter 4, titled "Shape FCM Algorithm", proposes another novel algorithm constructed based on the FCM algorithm with outstanding features of the Mapper algorithm. This

algorithm can exhibit the same clustering ability as the FCM algorithm and reveal some relationships by visualizing the global shape of data supplied by the Mapper algorithm. Four real-world datasets, including Unit Circle, Two Concentric Circles with Noise, 3D Trefoil Knot, and Reaven & Miller Diabetes, are implemented to demonstrate this algorithm's effectiveness. The experimental performances are analyzed and compared with the original method, the Mapper algorithm, and the fuzzy set-based improved method, the FM algorithm. The comparison is conducted concerning output visualization in the topological sense and clustering stability.

[C5] Chapter 5, titled "Conclusion", summarizes and evaluates the main thesis contributions based on the objectives identified at the outset. Thenceforth, the conclusions are drawn as a fulcrum for future research orientation and development.

1.4 Preliminaries

This section begins by briefly streamlining the theoretical frameworks in preparation for further study of the Mapper algorithm. Refer to popular classical textbooks on Algebraic Topology [30, 31], Computational Topology [11, 32], and Data Mining [33, 34] to fully understand these fundamentals.

1.4.1 Simplicial Complex

Definition 1 ([6]) Let X be the set of $k+1$ points $u_0, u_1, \ldots, u_k$ in the Euclidean space $\mathbb{R}^n$.

(a) A point x is called an *affine combination* of X if $x = \sum_{i=0}^{k} \lambda_i u_i$ with the real numbers λ_i satisfy $\sum_{i=0}^{k} \lambda_i = 1$.

(b) The set of all affine combinations of X is called its *affine hull*.

(c) An affine combination x of X is called its *convex combination* if the real numbers λ_i are all non-negative.

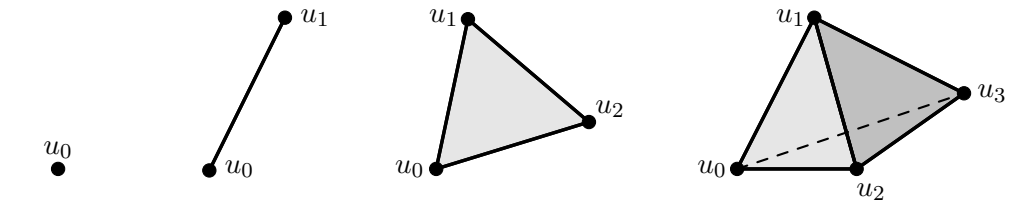
(d) The set of all convex combinations of X is called its *convex hull*.

(e) The system of points $u_0, u_1, \ldots, u_k$ is called *affinely independent* if the system of vectors $u_1 - u_0, u_2 - u_0, \ldots, u_k - u_0$ are linearly independent.

Definition 2 ([6]) Let X be the set of $k+1$ affinely independent points $u_0, u_1, \ldots, u_k$ in the Euclidean space $\mathbb{R}^n$.

(a) The convex hull of X is called a *k-simplex* or *simplex* with dimension k, spanned by X and denoted as $\sigma = [u_0, u_1, \ldots, u_k]$.

(b) Each element of X is called a *vertex* of the simplex σ.



(a) Point (b) Segment (c) Triangle (d) Tetrahedron

Figure 1: The basic simplices with dimensions from 0 to 4 from left to right.

(c) The simplex spanned by a subset of X is called a *face* of the simplex σ.

Figure 1 illustrates the basic simplices with dimensions from 0 to 4: 0-simplex is point, 1-simplex is segment, 2-simplex is triangle, and 3-simplex is tetrahedron.

Definition 3 ([6]) Let $K = \{S_1, S_2, \ldots, S_n\}$ be a finite collection of k simplices $S_1, S_2, \ldots, S_n$ in the Euclidean space $\mathbb{R}^n$.

(a) The collection K is a *geometric simplicial complex* if the following two conditions are satisfied:

- Every face of the simplex S_i is a simplex of K,
- Two arbitrary simplices S_i and S_j, either intersect to form the empty set or a common face.

(b) The *dimension* of the simplicial complex K is the maximum dimension of all its simplices $S_1, S_2, \ldots, S_n$.

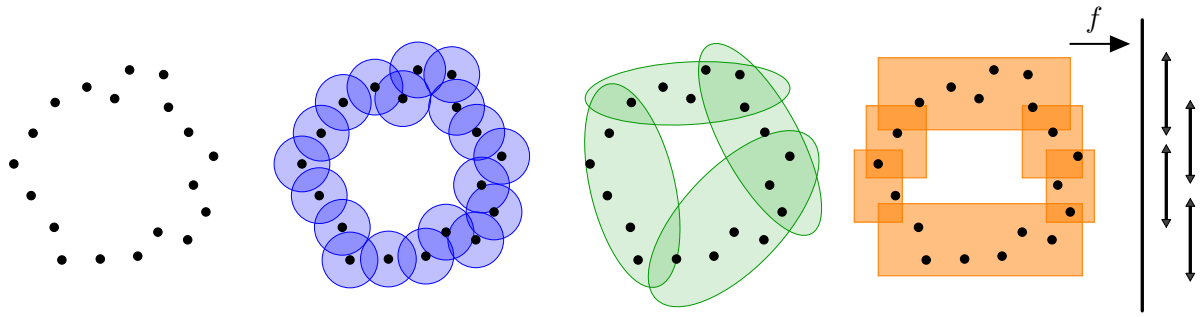
(c) The union $|K| = \bigcup_{i=1}^{n} S_i$ of all simplices $S_1, S_2, \ldots, S_n$ with the induced topology from $\mathbb{R}^n$ forms the *underlying space* of the simplicial complex K.

Definition 4 ([6]) Let V be a non-empty finite set and $\mathcal{K}$ is a collection of finite non-empty subsets of V.

(a) The collection $\mathcal{K}$ is an *abstract simplicial complex* if for all set $\sigma \in \mathcal{K}$, every non-empty subset of σ belongs to $\mathcal{K}$.

(b) Each element σ of $\mathcal{K}$ is its *simplex* and the cardinality of σ minus 1 is the *dimension* of the simplex σ.

(c) The dimension of the simplicial complex $\mathcal{K}$ is the maximum dimension of all its simplices $\sigma \in \mathcal{K}$.



(a) Point cloud (b) By open balls (c) By open sets (d) By preimage

Figure 2: The different cover methods on the same point cloud with the noisy circle structure.

The geometric form and abstract form of the simplicial complex have a close and unifying relationship. The abstract form $\mathcal{K}$ can be constructed from geometric form K by focusing only on the set of vertices of all its simplices and ignoring their geometric shapes. In this case, $\mathcal{K}$ is called a *vertex scheme* of K. In contrast, the geometric form is always associated with an abstract form $\mathcal{K}$. Specifically, it is possible to construct a geometric simplicial complex K in the Euclidean space $\mathbb{R}^{2d+1}$ from abstract simplicial complex $\mathcal{K}$ with dimension d [32, 6]. In this case, K is called a *geometric realization* of $\mathcal{K}$. It can be remarked that an abstract simplicial complex can be seen as a topological space and a geometric complex can be seen as a geometric realization of the underlying space [35, 36].

1.4.2 Cover and Nerve

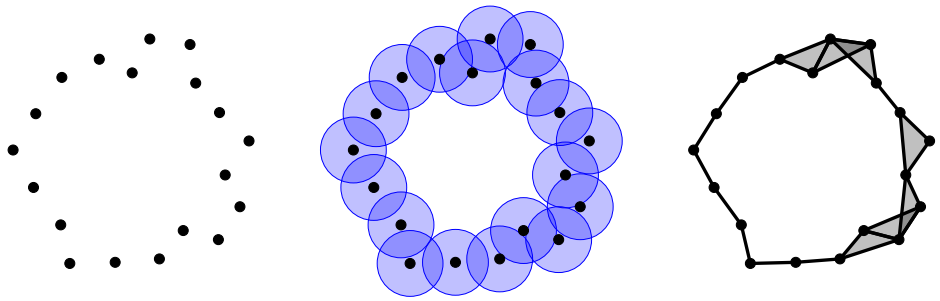
Definition 5 ([6]) For the topological space $\mathcal{X}$, a collection $\mathcal{U} = (U_i)_{i\in I}$ of open subsets $U_i \subseteq X$ is called an *open cover* of $\mathcal{X}$ if the condition $\mathcal{X} = \bigcup_{i\in I} U_i$ is satisfied.

Without special modification, all covers always contain a finite number of elements. Figure 2 illustrates three cover methods on a particular point cloud, including covering by open balls, arbitrary open sets, and the preimage of an available open cover on range through a continuous function.

Definition 6 ([6]) Let $\mathcal{X}$ be a topological space, $f : \mathcal{X} \to \mathbb{R}^n$ be a continuous function, and $\mathcal{U} = (U_i)_{i\in I}$ be an open cover of the Euclidean space $\mathbb{R}^n$.

(a) The collection of open sets $f^{-1}(\mathcal{U}) = (f^{-1}(U_i))_{i\in I}$ is called a *pull-back cover* of $\mathcal{X}$ induced by the pair $(f, \mathcal{U})$.

(b) The collection of all connected components obtained after decomposing each element $f^{-1}(U_i)$ of the pull-back cover $f^{-1}(\mathcal{U})$ is called a *refined pull-back cover*.



(a) Point cloud (b) Open cover (c) Nerve of cover

Figure 3: The nerve of an open cover on a sampled point cloud with the noisy circle structure.

Definition 7 ([6]) Let $\mathcal{U} = (U_i)_{i \in I}$ be an open cover of a topological space $\mathcal{X}$. The *nerve* of $\mathcal{U}$ is an abstract simplicial complex $N(\mathcal{U})$ with the vertex set $\mathcal{U}$ that is determined as follows:

$$N(\mathcal{U}) = \left\{ \{U_{i_0}, U_{i_1}, \ldots, U_{i_k}\} : \bigcap_{j=i_0}^{i_k} U_i \neq \emptyset \right\} \tag{1}$$

In the topological sense, a nerve is considered not only a discrete summary that captures exciting and valuable features obtained from the cover, but also ensures the preservation of the essential cover characteristics. Figure 3 illustrates a nerve of an open cover on a particular point cloud with the noisy circle structure.

The usefulness of the nerve in preserving the essential topological properties of the coating is confirmed against a robust theoretical framework, the *Nerve theorem*, as follows: If $\mathcal{U} = (U_i)_{i \in I}$ is an open cover of the subspace $\mathcal{X} \subseteq \mathbb{R}^n$ such that the intersection of any sub-collection of the U_i's is either empty or contractible, the nerve $N(\mathcal{U})$ and the subspace $\mathcal{X}$ are homotopy equivalent [6].

A cover satisfying the assumptions of the Nerve theorem is sometimes called a *good cover*. This theorem holds a crucial foundational role in computational topology and geometric inference [6]. It supplies an efficient method for encoding the homotopy type of continuous spaces into a simplicial complex with abstract combinatorial structures that describes the intersection pattern of a good cover [36]. Therefore, the topological characteristics of a continuous space can be studied on the nerve of its good cover. This idea is beneficial and perfectly suitable for efficiently designing data structures and algorithms [37].

1.4.3 Fuzzy C-Means Algorithm

Clustering is a process of dividing a set of objects into different groups, called clusters, so that the objects in the same group are more similar in some sense to each other than to those in other groups [34]. There are two types of clustering, *exclusive clustering*, and *overlapping*

Table 1: The description of the FCM algorithm.

Input	Dataset $\mathbb{X}$ with finite elements.
Parameters	– Number of clusters C, – Fuzzification exponent m, – Termination criteria: $\varepsilon \in (0;1)$ or $k_{\max}$.
Method	1: $k = 0$. 2: Initializing the fuzzy partition matrix $U^{(0)}$. 3: **repeat** 3.1: Calculating the cluster centroid matrix $\mathcal{C}^{(k)} = (v_j)_{1\times C}$ by using the following formula: $$\forall j = 1, 2, \ldots, C, v_j^{(k)} = \frac{\sum_{i=1}^{n} (u_{ij})^m x_i}{\sum_{i=1}^{n} (u_{ij})^m}.$$ 3.2: Updating the fuzzy partition matrix $U^{(k)} = (u_{ij})_{n\times C}$ by using the following formula: $$\forall i = 1, 2, \ldots, n, \forall j = 1, 2, \ldots, C, u_{ij}^{(k)} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|}\right)^{\frac{2}{m-1}}}.$$ 3.3: $k = k + 1$. 4: **until** $\max_{i,j} \left\{ \left| u_{ij}^{(k+1)} - u_{ij}^{(k)} \right| \right\} < \varepsilon$ or $k = k_{\max}$.
Output	– Cluster centroid matrix $C = [v_j]$, – Fuzzy partition matrix $U = [u_{ij}]$.

clustering. In exclusive clustering, each object belongs to exactly one cluster. In overlapping clustering, each object can belong to two or more clusters depending on the membership function. Membership degree, assigned to each object, indicates the degree to which this object belongs to each cluster. The fuzzy set concept introduced by Zadeh in 1965 [38] is considered the inspiration for overlapping clustering. Fuzzy set theory has become an increasingly useful tool to describe situations in which the data are imprecise or vague, such as linguistics [39, 40], decision-making [41, 42], web mining [43, 44], frequent itemset mining [45, 46], bioinformatics [47, 48, 49], and so on.

In many situations, overlapping clustering is more natural than exclusive clustering. This method randomly separates data points into clusters with different overlapping percentages. It is pretty compatible with the cover choice step for the filter range in the Mapper algorithm. According to the topological meaning, these clusters serve as a cover of the point cloud space.

While the K-Means algorithm, also referred to by Hard C-Means, is a popular method for exclusive clustering, then the FCM algorithm is an essential representative for overlapping

clustering. This algorithm was first introduced in 1981 [50] by Bezdek, based on improving on earlier clustering method in the excellent monograph produced in 1973 [51] by Dunn.

Given a dataset with finite elements $\mathbb{X} = \{x_1, x_2, \ldots, x_n\}$, the FCM algorithm organizes data points into the C clusters characterized by some centroids $\mathcal{C} = \{v_1, v_2, \ldots, v_C\}$. The main goal of this algorithm is to iteratively optimize (minimize) the following *objective function*:

$$\sum_{i=1}^{n}\sum_{j=1}^{C}(u_{ij})^m \left\|x_i - v_j\right\|^2, \tag{2}$$

where the *membership degree* u_{ij} of the point x_i in the cluster j is constrained in the following way:

$$\forall i = 1, 2, \ldots, n, \forall j = 1, 2, \ldots, C, u_{ij} \in [0; 1], \tag{3}$$

and

$$\forall i = 1, 2, \ldots, n, \sum_{j=1}^{C} u_{ij} = 1. \tag{4}$$

In Equation 2, the *fuzzification exponent* m is any real number greater than 1, and $\|.\|$ is the *Euclidean metric* expressing dissimilarity between arbitrary points and a given centroid.

The fuzzification exponent m plays an essential role in this algorithm. The best choice for this exponent is made experimentally [52]. For most data, the value of m is indicated to be the best lies within $[1.5; 2.5]$, the best deal is obtained by evaluating some cluster validity indices [53]. Based on empirical studies, most researchers have assigned this parameter to a fixed value, $m = 2.0$. In 2008, Pedrycz and Oliveira [54] gave experimental evidence behind the selection of the fuzzification exponent at this value. The optimal values of the fuzzification exponent are typically lower than the commonly used value of 2.0. However, in this study, the value of the fuzzification exponent m is set as a constant 2.0.

The fuzzy partition is carried out through iterative minimization of the objective function through updating the *cluster centroids* v_j and the *partition matrix*, generated by membership degrees, u_{ij} according to the following formulas:

$$\forall j = 1, 2, \ldots, C, v_j = \frac{\sum\limits_{i=1}^{n}(u_{ij})^m x_i}{\sum\limits_{i=1}^{n}(u_{ij})^m}, \tag{5}$$

and

$$\forall i = 1, 2, \ldots, n, \forall j = 1, 2, \ldots, C, u_{ij} = \frac{1}{\sum\limits_{k=1}^{C}\left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|}\right)^{\frac{2}{m-1}}}. \tag{6}$$

The iterative algorithm terminates after k iterations when one of the following two termi-

nation conditions have been satisfied:

$$\max_{i,j} \left\{ \left| u_{ij}^{(k+1)} - u_{ij}^{(k)} \right| \right\} < \varepsilon \in (0;1)\,, \tag{7}$$

or

$$k = k_{\max}, \tag{8}$$

where ε is a given positive real number in the unit interval and $k_{\max}$ is a given positive integer corresponding to the number of iterations. The FCM algorithm is described clearly and precisely in Table 1.

1.5 Discussion

By introducing basic contours on TDA literature, this chapter has clarified the motivations of the thesis in using fuzzy theory to approach TDA. Its goals have been explicitly determined for studying in the following chapters based on these orientations. This thesis aims are associated with the detailed contents of its three major chapters. In addition, this chapter also presents concisely the fundamental theory required for TDA and fuzzy clustering to serve as the framework for understanding the expertise offered in the following chapters. More complete insights into these can be found in the classic textbook on algebraic topology and data mining.

2 Mapper Algorithm

Topological tools are becoming more powerful and dominant in data analysis because of their outstanding characteristics such as coordinate independence, invariance under transformation, and meaningful compressed representation. These characteristics make TDA a promising area of research composed of two fundamental streams, including persistent homology and the Mapper algorithm. While persistent homology is strong regarding small perturbations in data and provides a compact representation for studying its qualitative features with complex structure, the Mapper algorithm can summarize meaningful insights and valuable topological information from the high-dimensional point cloud. This algorithm increasingly thrives on a stabilized theoretical foundation with practical applications and diverse, intuitive, user-friendly implementations. Its development has inspired systematically reviewing it in terms of frameworks and applications. Chapter 2 reviews the Mapper algorithm with specific descriptions and intuitive, easy-to-understand illustrations. Its variations and applications are also presented over time, systematically and thoroughly. We also briefly introduce the popular available packages that activate this algorithm as the core of their operations. Furthermore, we also discuss its current limitations to guide future research and development. It is the first time the Mapper algorithm has been independently reviewed as an outstanding representative of TDA to the best of our knowledge.

Overall, the contributions of Chapter 2 can be summarized as follows:

[C2.1] Describing the Mapper algorithm carefully with intuitive and easy-to-understand illustrations.

[C2.2] Systematizing the variations of the Mapper algorithm thoroughly over time and discussing well-known applications of this algorithm in emerging fields systematically and clearly.

[C2.3] Providing a concise and complete introduction to the available tools using the Mapper algorithm as a theoretical framework.

[C2.4] Presenting the current limitations and future directions for the research to develop the Mapper algorithm continuously.

Moreover, the rest of this chapter is organized as follows. Section 2.1 condenses on the brief frameworks of computational topology for understanding the Mapper algorithm, which is introduced with its variants logically and clearly in Section 2.2. Section 2.3 is dedicated to discussing the algorithm's applications in science and practice. Section 2.4 provides an introduction to the available tools for this algorithm on multiple platforms. Section 2.5 presents the current limitations and Section 2.6 guides future research to develop this algorithm. Finally, Section 2.7 marks the end of this review on the Mapper algorithm, one of the most promising representatives of TDA.

Table 2: The description of the Mapper algorithm.

Input	Dataset $\mathbb{X}$ with finite elements.
Parameters	– Distance metric d, – Filter function $f : \mathbb{X} \longrightarrow \mathbb{R}$, – Cover $\mathcal{I}$ of the filter range $f(\mathbb{X})$ with N regular intervals and the same overlapping percentage p, – Clustering algorithm $\mathcal{C}$ (option).
Method	1: Projecting all data points in $\mathbb{X}$ to $\mathbb{R}$ by using the filter f. 2: Covering $f(\mathbb{X})$ by N regular intervals and the same overlapping percentage p between adjacent intervals. 3: Decomposing the pre-image $f^{-1}(I_i)$ of each cover interval I_i into clusters $C_{i,1}$, $C_{i,2}$, $C_{i,3}$, $\ldots$, C_{i,n_i} by using the clustering algorithm $\mathcal{C}$ or partitioning connected components based on the distance metric d. 4: Constructing the simplicial complex $\mathcal{G}$ defined by clusters C_{ij} and their intersections.
Output	Simplicial complex $\mathcal{G}$ as a geometric representation of $\mathbb{X}$.

2.1 Description

This algorithm takes *input* as a discrete dataset $\mathbb{X}$ and starts working with four user-defined *parameters* as follows:

– Distance metric d,

– Filter function $f : \mathbb{X} \to \mathbb{R}$,

– Cover $\mathcal{I}$ of the filter range $f(\mathbb{X})$ with the N regular intervals and the same overlapping percentage p between adjacent intervals,

– Clustering algorithm $\mathcal{C}$.

The input data is combined with user parameters to trigger the algorithm operation through the following steps:

(1) **Step 1.** Projecting all points in the dataset $\mathbb{X}$ to the real line $\mathbb{R}$ by using the filter function f.

– Computing the filter values $f(x_i)$ for all $x_i \in \mathbb{X}$.

– Determining the filter range $f(X) = [f_{\min}; f_{\max}]$ where

$$f_{\min} = \min_{x_i \in \mathbb{X}} f(x_i), \tag{9}$$

and

$$f_{\max} = \max_{x_i \in \mathbb{X}} f(x_i). \tag{10}$$

(2) **Step 2.** Covering the filter range $f(\mathbb{X})$ by the cover $\mathcal{I}$ including N regular intervals and the same overlapping percentage p between adjacent intervals.

- Determining the length l of cover intervals as follows:

$$l = \frac{f_{\max} - f_{\min}}{(N-1)(1-p)+1}. \tag{11}$$

- Determining the regular cover intervals $I_i = [a_i; b_i]$ for all $i = 1, 2, 3, \ldots, N$ with the same overlapping percentage p as follows:

$$a_1 = f_{\min}, \tag{12}$$

$$\forall i = 2, 3, 4, \ldots, N, a_i = a_1 + (i-1)(1-p)\, l, \tag{13}$$

$$\forall i = 1, 2, 3, \ldots, N-1, b_i = a_i + l, \tag{14}$$

$$b_N = f_{\max}. \tag{15}$$

- Determining the cover $\mathcal{I} = \{I_1, I_2, I_3, \ldots, I_N\}$ on the filter range $f(\mathbb{X})$.

(3) **Step 3.** Decomposing the pre-image $f^{-1}(I_i)$ of each cover interval I_i into clusters $C_{i,1}$, $C_{i,2}$, $C_{i,3}$, …, C_{i,n_i} by using the clustering algorithm $\mathcal{C}$ or partitioning connected components based on the distance metric d.

- Determining the pre-image X_i of each cover interval I_i as follows:

$$X_i = f^{-1}(I_i) = \{x_k \in \mathbb{X} : a_i \leq f(x_k) \leq b_i\}. \tag{16}$$

- Applying the clustering algorithm $\mathcal{C}$ or partitioning connected components for each pre-image X_i to separate into clusters $C_{i,1}$, $C_{i,2}$, $C_{i,3}$, …, C_{i,n_i} as follows:

$$\forall i = 1, 2, 3, \ldots, N, X_i = \bigcup_{j=1}^{n_i} C_{i,j}, \tag{17}$$

$$\forall i = 1, 2, 3, \ldots, N, \forall j, k = 1, 2, 3, \ldots, n_i, j \neq k \Longrightarrow C_{i,j} \cap C_{i,k} \neq \emptyset. \tag{18}$$

- Determining the refined pull-back cover $\mathcal{C}$ of $\mathbb{X}$ as follows:

$$\mathcal{U} = \{C_{1,1}, C_{1,2}, \ldots, C_{1,n_1}, C_{2,1}, C_{2,2}, \ldots, C_{2,n_1}, \ldots, C_{N,1}, C_{N,2}, \ldots, C_{N,n_N}\}. \tag{19}$$

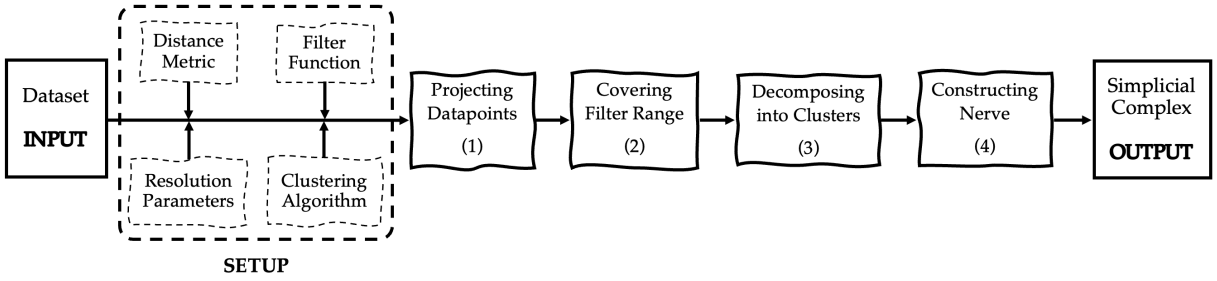


Figure 4: The schematic diagram of the Mapper algorithm.

(4) **Step 4.** Constructing the simplicial complex $\mathcal{G}$ defined by clusters C_{ij} and their intersections.

- Determining the nerve $\mathcal{G} = (V, E)$ of the refined pull-back cover $\mathcal{U}$ of $\mathbb{X}$ based on the clusters as follows:
  - Each node $v_{i,j}$ corresponds to the cluster $C_{i,j}$,

$$V = \{v_{1,1}, v_{1,2}, \ldots, v_{1,n_1}, v_{2,1}, v_{2,2}, \ldots, v_{2,n_1}, \ldots, v_{N,1}, v_{N,2}, \ldots, v_{N,n_N}\} \tag{20}$$

  - Each edge between $v_{i,j}$ and $v_{i+1,k}$ is constructed if and only if

$$C_{i,j} \cap C_{i+1,k} \neq \emptyset \tag{21}$$

- Visualizing the nerve $\mathcal{G}$ with the nodes' characteristics, color and size.
  - The color of a node often represents the average of filter values at points in the corresponding cluster. Usually, red indicates a maximal value and blue indicates a minimal value. The colors ranging from red to blue express the colored values ranging from high to low.
  - The size of a node represents the number of points in the corresponding cluster.

The *output* of this algorithm is a network graph or a simplicial complex in general that is a *geometric representation* of the previous original data. It should be noted that when the filter is a real-valued function, the simplicial complex is constructed with a dimension not greater than one. Therefore, to generate a simplicial complex with multiple dimensions, the Mapper users must use more than one filter with real value or one filter with vector value. This algorithm is briefly described in Table 2, Figure 4, and clearly illustrated in Figure 5.

The *metric* is sometimes not considered a parameter because it is only used to calculate distances between data points for clustering or partitioning. The *clustering algorithm* is essentially the discrete version of identifying the connected components of a topological space. This algorithm is an optional parameter because there are two strategies to separate the pre-image of each cover interval into subgroups, including clustering the data points or partitioning con-

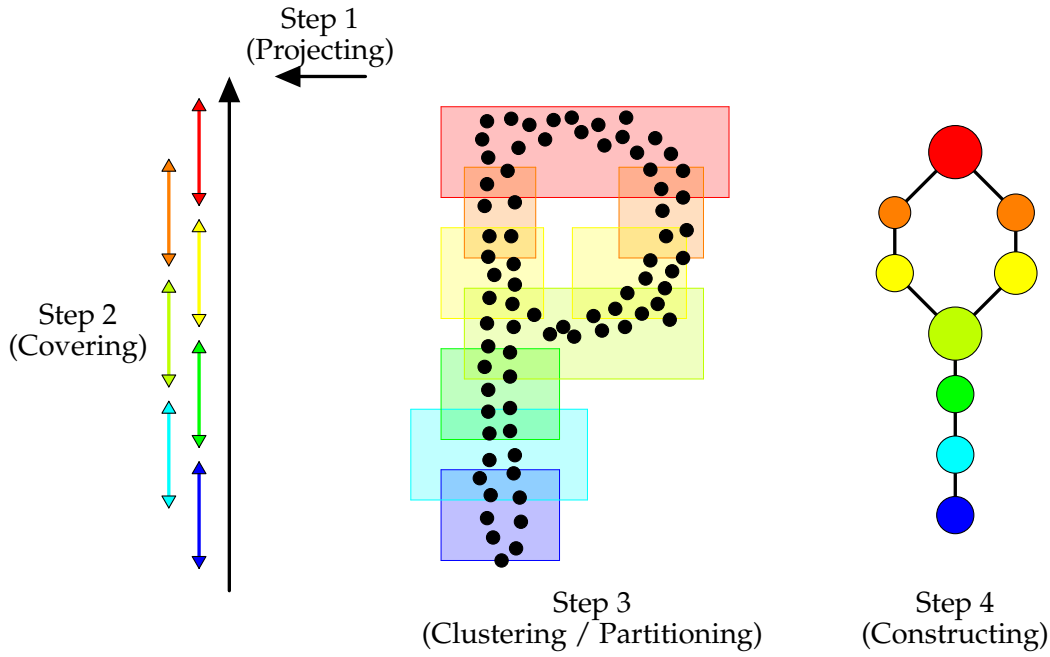


Figure 5: The implementation of the Mapper algorithm on a point cloud with a noisy P structure.

nected components of a built neighboring graph [13]. They are responsible for converting from the topological representation to the statistical one to facilitate visualization [35]. It should be noted that the Mapper algorithm is not affected by a particular clustering algorithm [36].

The *filter function* can be seen as forgetting the coordinates, but most useful functions are statistically more meaningful. There is no special rule in choosing a filter for the Mapper algorithm in the literature. It is also said to depend on the context and characteristics of data [55]. The classical choices in well-known works of this algorithm can be mentioned as KDE [13, 56], eccentricity [17], graph Laplacians [13], L-infinity centrality [17, 57, 58], SVD [17, 57, 58], PCA [58, 59, 60, 61, 26, 62]], multi-dimensional scaling [25], distance functions [63, 26], t-SNE [21], UMAP [26], and so on. The output network is quite sensitive to this parameter. Figure 6 illustrates how the filter function affects the results of this algorithm. Perpendicular to the horizontal and vertical axes, these two functions are sequentially used as filters on the same sampled point cloud with the noisy P structure. The visualizations obtained for these two cases are very different.

The filter range cover is characterized by two factors, called the *resolution parameters*, including the number of regular intervals and the overlapping percentage between adjacent intervals. These factors significantly affect the shape of the generated simplicial complex. The number of intervals positively correlates with the number of nodes, while the overlapping percentage positively correlates with the connectivity between nodes. As the number of intervals changes, the number of nodes also changes in the same direction. The same goes for the overlapping

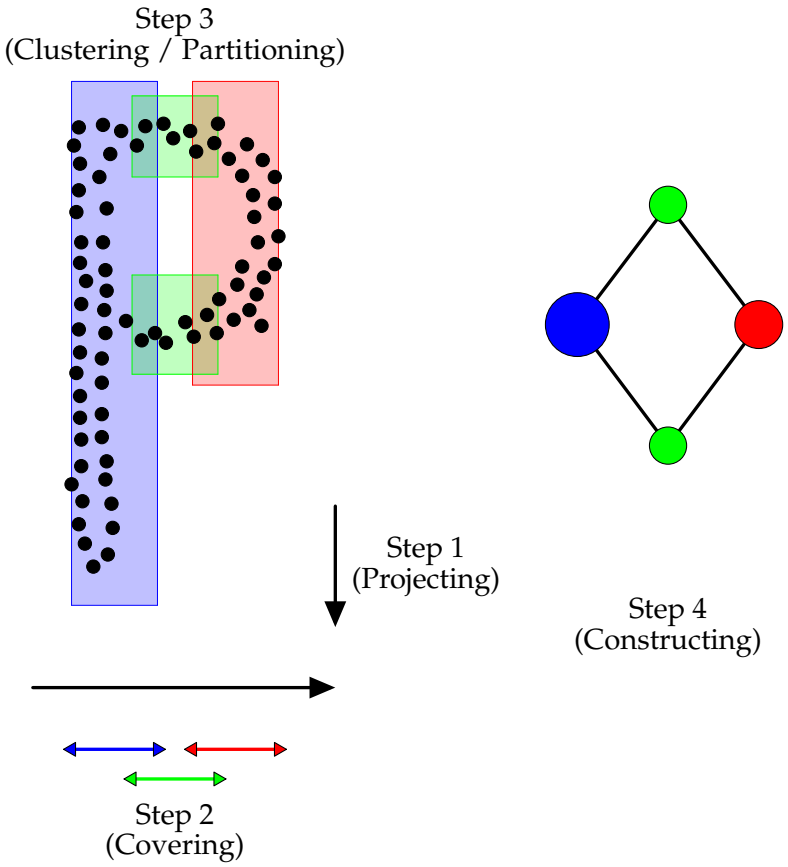


Figure 6: The implementation of the Mapper algorithm on a point cloud has a noisy P structure when changing the filter function compared to Figure 5.

percentage and the network connectivity between nodes. The choice of these parameters is entirely up to the user's opinion based on the graphical simplicial complex at the output. It is worth noting that the selecting the resolution parameters significantly affects the Mapper output. This algorithm is often unstable because a slight change in the cover on the filter range can lead to a substantial change in the generated network [35]. Indeed, Figure 6 illustrates the influence of resolution parameters on the resulting shape. Two cover methods on the filter range, 7 intervals, 40% overlap in Figure 5 and 3 intervals, 20% overlap in Figure 7, are also used sequentially as resolutions on the same sampled point cloud with the noisy P structure. The visualizations obtained for these two cases are also entirely different.

2.2 Variation

The Mapper algorithm was invented in 2007 [13], but its emergence in the scientific community was highlighted in 2013 [17]. Since then, it has increasingly demonstrated its effectiveness and strength in many fields. In addition to combining with other analytical tools to achieve high efficiency in data analysis, many of its variations have been conceived for particular pur-

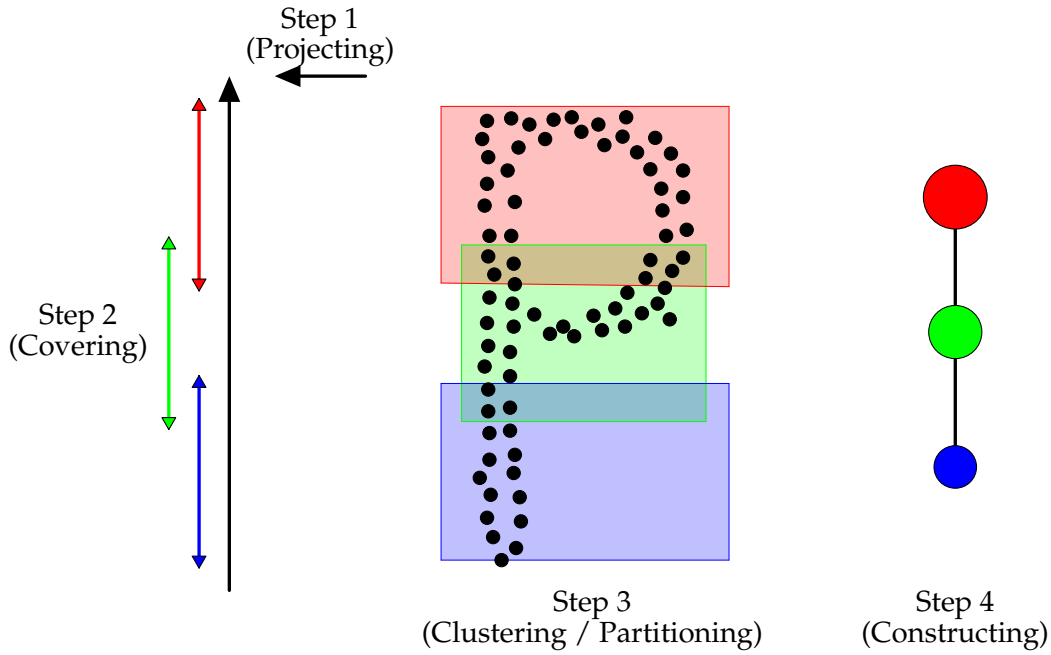


Figure 7: The implementation of the Mapper algorithm on a point cloud has a noisy P structure when changing the resolution parameters compared to Figure 5

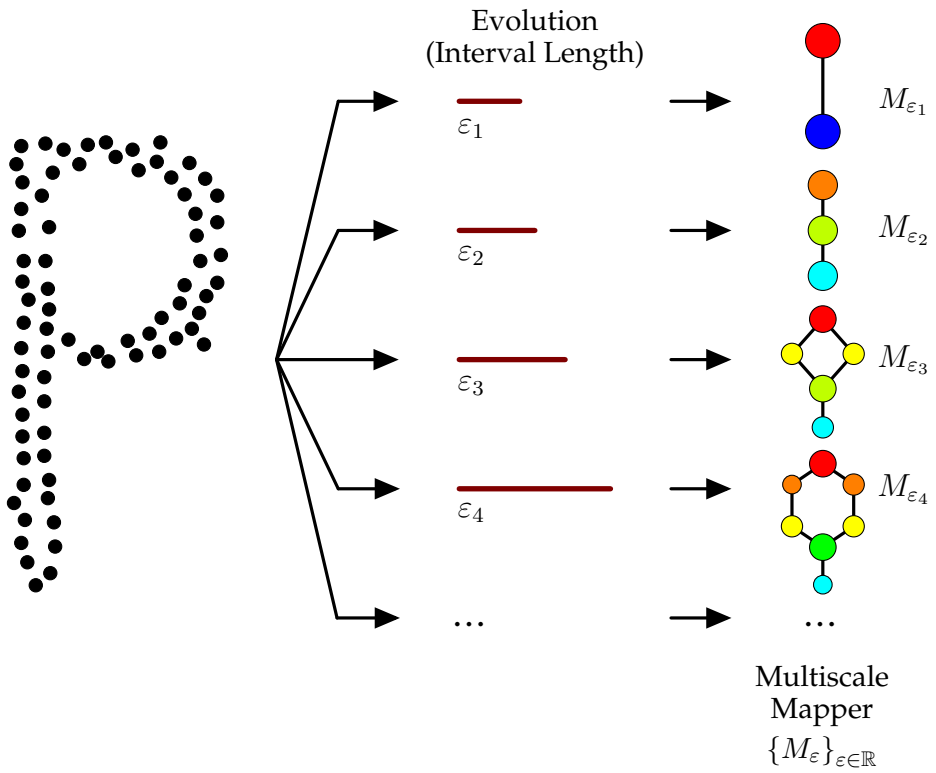


Figure 8: The implementation of the MM algorithm on a point cloud with a noisy P structure.

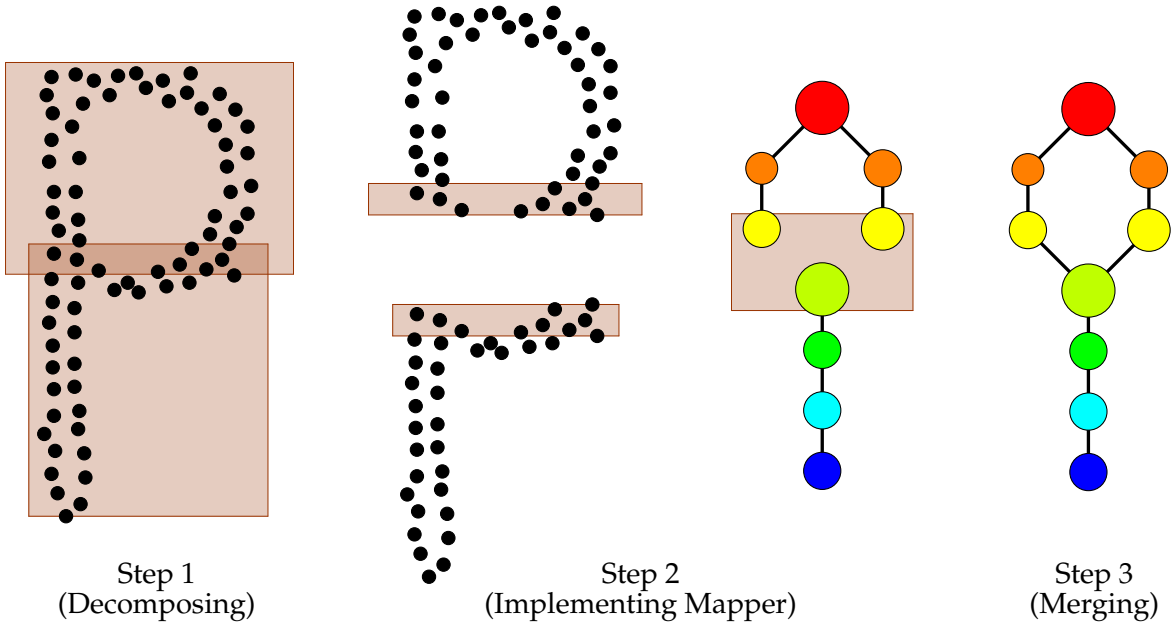


Figure 9: The implementation of the PM algorithm on a point cloud with a noisy P structure.

poses during its development. This subsection presents named variants based on the theoretical framework of the Mapper algorithm, including Multiscale Mapper [64, 65], Parallel Mapper [66], Mapper on Graph [67], Ball Mapper [68], Fuzzy Mapper [35], Shape Fuzzy C-Means [37], Ensemble Mapper [69], and Deep Graph Mapper [70].

The *Multiscale Mapper* algorithm was proposed in 2016 [64] as an evolution of the Mapper method inspired by looking at data through a tower of covers instead of a fixed cover on the filter range. This tower creates a building of simplicial complexes linked together by simplicial maps. Through this approach, the authors have studied the results obtained from the algorithm through its structure and stability. It should be emphasized that stability is a highly desirable characteristic because of its ability to imply robustness to noise in data and measurements. The MM algorithm is proven to satisfy the stability property by the homology theory, while the Mapper algorithm does not. Besides, this stability promotes the design of efficient algorithms for their formation and approximation in practice. With a simplicial complex and a real-valued piecewise-linear function, this study shows that the persistence diagram can be computed precisely from only the 1-skeleton of the original complex. More broadly, for a simplicial complex and a general function, an even simpler combinatorial version of the MM algorithm acts only on vertex sets of the complex with connectivity given by its 1-skeleton graph. It approximates the exact persistence diagram thanks to stability based on a "goodness" condition on the tower of covers. Furthermore, topological information analysis of the MM outputs was given in comparative correlation with the Nerves, the Reeb Spaces, and the Mapper outputs to better understand these structures and facilitate their practical usage [65]. This algorithm is illustrated

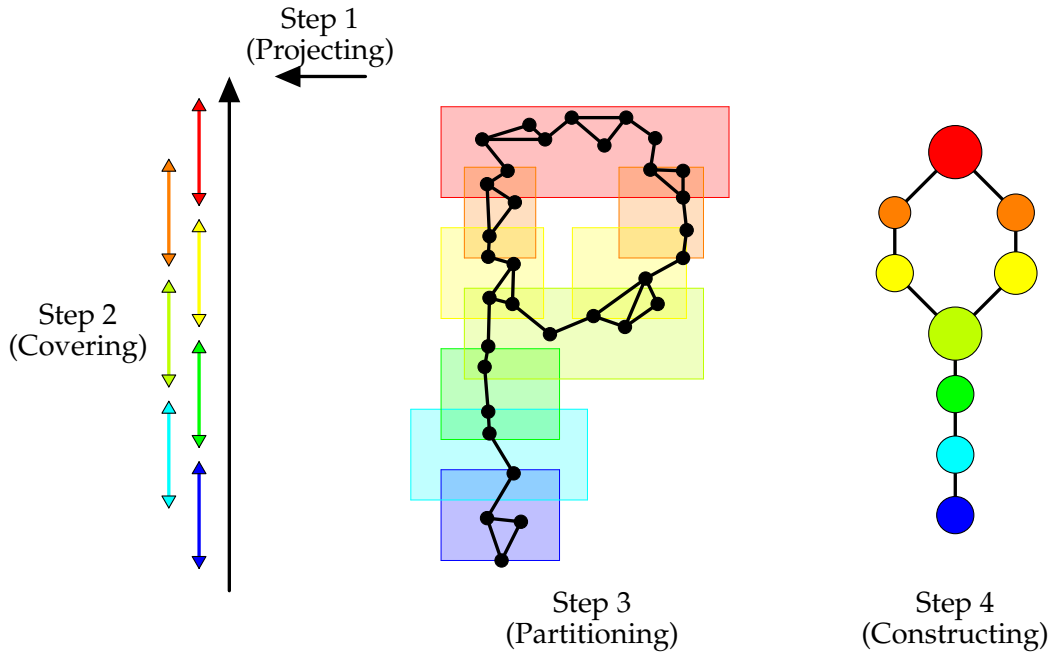


Figure 10: The implementation of the MOG algorithm on a graph with a noisy P structure.

visually in Figure 8.

The *Parallel Mapper* algorithm was proposed in 2017 but officially announced in 2020 [66] as a Mapper distribution on a set of processors running in parallel. It works based on a divide and conquer strategy for the codomain cover. Its efficiency is also demonstrated and reported clearly through the performance experiments. The highlight of the PM algorithm is reflected in the ability to contribute the Mapper algorithm with multi-resolutions. This property is useful for interactive applications where it is possible to increase the resolution for some meaningful data subgroup with a specific purpose. Figure 9 illustrates the operation steps of this algorithm clearly.

The *Mapper on Graph* algorithm first appeared in 2018 [67] as a variation of the Mapper method. It is constructed upon a popular tool in TDA, which provides a robust theoretical foundation for summarizing the valuable information of network data and preserving their core structures. This algorithm is developed to target weighted and undirected graphs to generate characteristic-preserving summaries by transforming large graphs into hierarchical representations. Accompanied by the algorithm, an official distribution [71] in Python under the GNU General Public License is also published to allow users to enable interactive explorations for their data. The effectiveness of this approach is demonstrated by evaluating experiments on both synthetic and real-world data. It can be said that the MOG algorithm emerges as a new impetus to leverage TDA tools in addressing the challenges of visualizing and mining on graph data. The operation steps of this algorithm are illustrated visually in Figure 10.

Inspired by the Mapper method, the *Ball Mapper* algorithm is described in 2019 [68] as an

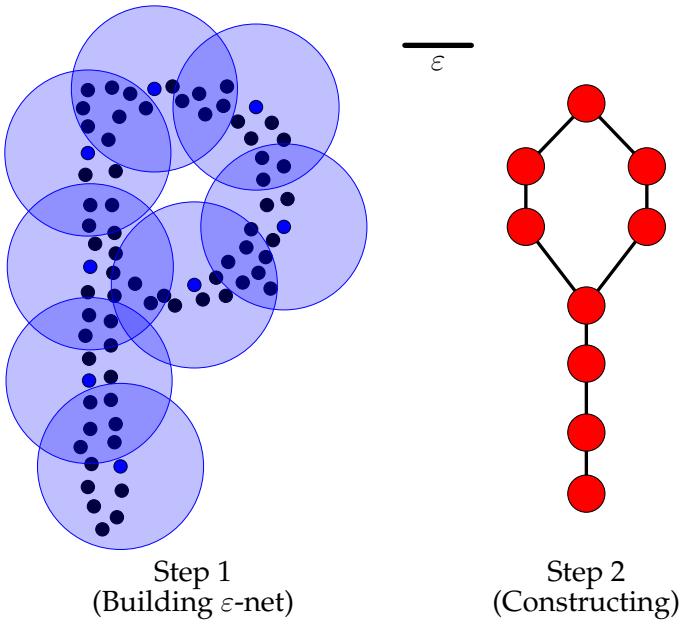


Figure 11: The implementation of the BM algorithm on a point cloud with a noisy P structure.

effective tool in exploratory data analysis by tightly encapsulating both the local and global structure of the high-dimensional datasets. This algorithm, with simple construction, is easy to compute and analyze, and extend. It is proposed to cover the point cloud with balls with equal radius instead of a refined pull-back cover generated from the filter function and a cover on the filter range. It has a straightforward description with steps that are not too difficult to understand. Its complexity is also discussed with explicit calculations. By analyzing the relationship between two algorithms, the BM algorithm is recommended as a complementary technique used with the Mapper algorithm in data analysis. This algorithm has been published with free official distribution [72] in R under the MIT License. Although published as a pre-print, the author has also managed to demonstrate its effectiveness when applying it to analyze some specific data, including visualizing financial ratios as an abstract two-dimensional graph [73], examining the economic topology of the Brexit vote [74], analyzing and evaluating macroeconomy [75], visualizing the Covid-19 evolution [76], refining the understanding of corporate failure [77], and applying to knot theory concerning the Mapper algorithm [78]. An illustration of this algorithm can be found in Figure 11.

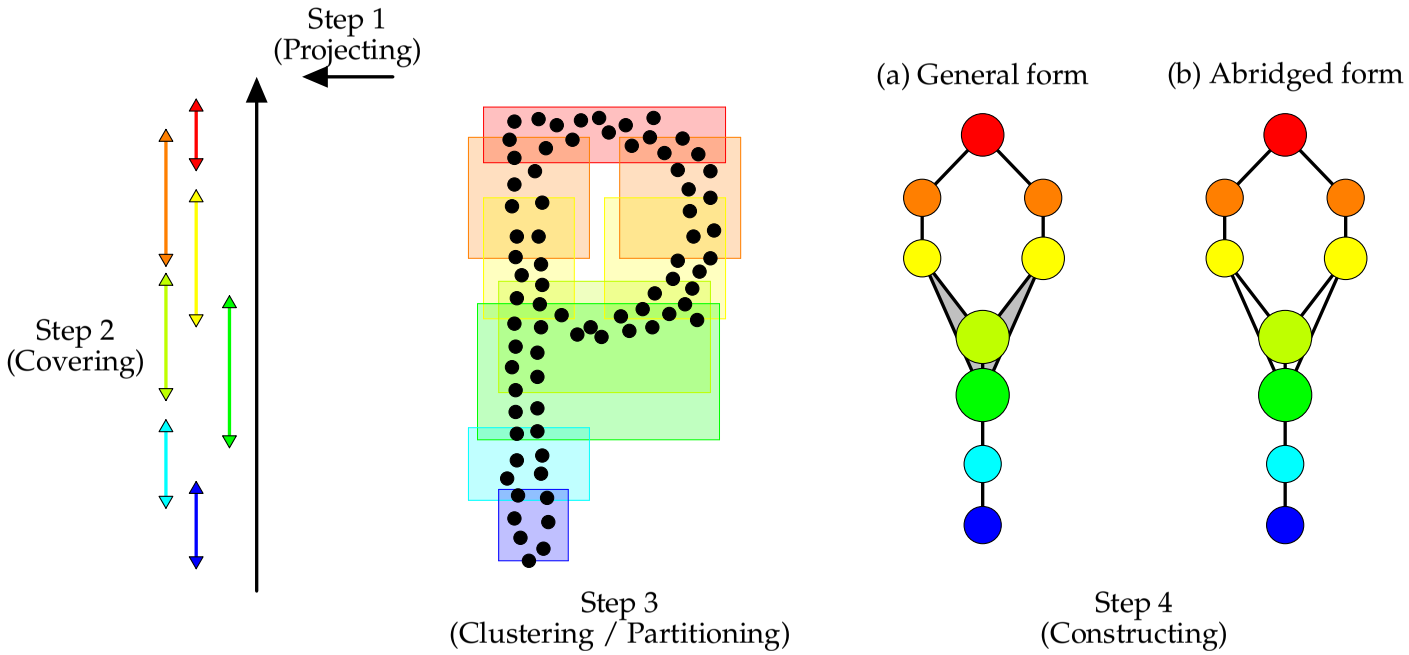


Figure 12: The implementation of the FM algorithm on a point cloud with a noisy P structure.

The *Fuzzy Mapper* algorithm was proposed in 2020 [35] to improve the resolution parameters for the Mapper method by choosing a filter cover naturally by overlapping clustering features. The FCM [51, 50], an intelligent technique of overlapping clustering, is used to automate the division of cover intervals with a random overlapping percentage. The new algorithm has been appreciated for its ability to generate graphs like those of the Mapper algorithm from a topological standpoint, although their cover choices are different. Its effectiveness is also demonstrated by evaluating experimental results on the popular real-world datasets mentioned earlier. The good clustering ability of the FM algorithm stems from the reasonable division of the cover intervals over filter range based on the density distribution of the data points. A visual illustration of this algorithm is shown in Figure 12.

The *Shape Fuzzy C-Means* algorithm can also be considered a variant of the Mapper method in 2021 [37] when constructed on the FCM algorithm framework and equipped with special characters in global shape detection. It demonstrates fuzzy clustering like the FCM algorithm and creates a concise, realistic, intuitive topological summary for high-dimensional datasets like the Mapper algorithm. Combining two capabilities, clustering and shape detection, is a groundbreaking, logical and feasible idea. This combination is effective for both algorithms. The FCM algorithm is more prominent in new outstanding capabilities in simplifying and visualizing data with qualitative analysis. The Mapper algorithm is also simplified in choosing parameters to generate the most informative presentations. This work has also shown the SFCM algorithm's effectiveness in fuzzy clustering ability and structure detection through experiments on high-dimensional real-world datasets. Furthermore, the SFCM algorithm is also a particular enhanced case of the FM algorithm in a certain sense. It is briefly and intuitively described in Figure 13.

The *Ensemble Mapper* algorithm is the latest version that optimizes resolution parameters, including the number of intervals and the overlapping percentage. It has just been conceived in 2021 [69] on the theoretical foundation of the Mapper method but optimizes the choice of resolution parameters according to the ensemble technique [79]. Ensemble methods are practical tools in improving the robustness, stability, and accuracy of clustering solutions and developed to combine clustering results rather than find the best one [80]. The inspired idea to create this algorithm is also quite simple as the fuzzy clustering ensemble method is applied to combine the results generated by the Mapper algorithm under different resolution parameter values. Developing a combined effect based on an ensemble of the various base Mapper results shows that it is unnecessary to find the optimal deals for resolution parameters when using the Mapper algorithm. Simultaneously, the EM algorithm outperforms the existing algorithms in visualization and numeric scores by experiments on real-world datasets. A model of this algorithm is presented in Figure 14.

The *Deep Graph Mapper* algorithm was first mentioned in 2020 and officially in 2021 [70] as a topologically-grounded method that merges the Mapper method with GNN to generate informative hierarchical graph visualizations. These visualizations not only aid in discerning

complex graph structures, but also provide a practical means to understand the models used to solve various tasks. In this work, the Mapper algorithm is rigorously proven by mathematical reasoning as a generalization of soft cluster assignment pooling methods, including Diff-pool [81] and minCUT [82]. This evidence provides an efficient connection between graph pooling and TDA. Based on this connection, a simple Mapper-based PageRank pooling operator is proposed with the ability to achieve similar or superior results compared to state-of-the-art methods on several graph classification benchmarks. An official code [83] was also published in Python for this work with the promise of making the DGM algorithm a complete graph visualization library based on the Mapper algorithm. It is easy to see that the DGM algorithm is a case of the MOG algorithm when effectively selecting the filter function according to the deep learning orientation.

Besides variations that are formed directly from the core algorithm, several other algorithms are also constructed upon it as an intermediate step or stage for processing data, such as *Two-Tier Mapper* [84], *Mapper-Induced Manifold Alignment* [85], *Mapper Based Classifier* [86], etc. The TTM algorithm was created in 2017 and officially published in 2019 [84] as an un-biased topology-based clustering method for enhanced global gene expression analysis. The impetus for its formation is derived from using TDA to overcome the existing disadvantages of current clustering methods, including dependence on user-defined parameters, lack of stability, and inadequacy for small data sets. The TTM algorithm is expected to become a promising tool for personalized medicine with outstanding advantages. This algorithm is published with a free R library [87] under the GNU General Public License in Bioconductor that allows its widespread adoption. The MIMA algorithm is proposed on the foundation of merging SSMA with the Mapper algorithm [85]. It is the first time a multi-modal data fusion algorithm is applied to fuse optical data, and synthetic aperture radar data and TDA are involved in remote sensing. The SSMA method usually creates a topological structure using K-nearest neighbor, while the MIMA algorithm implements the Mapper method to generate a novel topological structure through spectral clustering in a data-driven fashion. The experimental results have indicated the superior performance of the MIMA algorithm when fusing data related to land cover land use classification and local climate zone classification. The MBC method is described as a Mapper-based classifier to project data onto a latent space. This latent space is obtained using advanced techniques, such as PCA or AE. Notably, this classifier method is immune to any gradient-based attack and improves robustness over traditional convolutional neural networks [86]. New algorithms created on the framework of the Mapper algorithm have all proven their effectiveness for a specific need in the data processing. It partly demonstrates its superiority as an outstanding principal representative of TDA.

2.3 Application

The Mapper algorithm was proposed as a new method to analyze qualitatively, succinctly simplify, and effectively visualize high-dimensional datasets [13]. Its motivation comes from

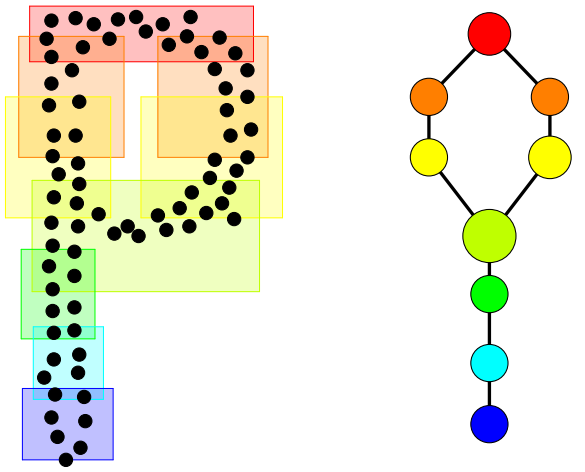


Figure 13: The implementation of the SFCM algorithm on a point cloud with a noisy P structure.

the impossibility of structural visualization and discernment for massive datasets, even with widespread low-dimensional projections. It is proposed to transform massive datasets with high dimensionality into visual simplicial complexes with far fewer vertices to capture useful geometric and topological information obscured by their immense size, corresponding to a specified resolution. It resembles the Reeb graph but is more general and adaptable to even higher dimensional objects.

2.3.1 Data Clustering

The most popular real-world application of the Mapper algorithm is its clustering capability that reveals data insights with a simple, intuitive, and helpful summary. This algorithm's ability was highlighted in the first appearance with experiments on the Reaven and Miller Diabetes dataset [88, 13]. It has been demonstrated as an automated tool for discovering the boomerang phenomenon with two floppy wings and one fat middle, even when the best possible projections do not generate such a good image. The respective patients in each wing were considered to have fundamentally different diseases, which correspond to the division of diabetes into the adult-onset and juvenile-onset forms. This result has great significance based on the famous previous study conducted by Reaven and Miller [88].

In 2009, this algorithm was also chosen as a computational approach in analyzing simulation data for biomolecular folding pathways [56]. Its success is shown in structural evidence from computer simulations supporting that RNA hairpin folding has two dominant pathways with many intermediate states and is kinetically separated. In addition, the Mapper algorithm, with its optimization of density filters and clustering schemes, is seen as a promising way to

solve distribution heterogeneity, deal with multiple pathway data, and be less sensitive to metric choice.

This algorithm activates as the second component of Progression Analysis of Disease to provide a helpful summary that can be used to further explore high-throughput biological data [89]. With its outstanding ability, the Mapper algorithm has identified a unique subgroup of Estrogen Receptor-positive breast cancers, in which their levels of c-MYB are high, and their levels of innate inflammatory genes are low. The vital surprise is that the patients in this group exhibit complete survival and no metastasis. This group has a clear and distinct molecular signature with statistical significance and highlights coherent biology but is not visible for the clustering methods.

It can be said that the ability to cluster data and the Mapper algorithm also proved effective in identifying a meaningful subgroup quickly that traditional clustering methodologies fail to find. This affirmation is again demonstrated in the famous work [17], which marks the scientific community's widespread interest in this method. This work has also shown that it can handle many different data types by using three datasets: gene expression from breast tumors, voting data from the United States House of Representatives, and player performance data from the National Basketball Association.

The characteristic of detecting significant subgroups is an essential highlight for clustering ability that traditional methods of the Mapper algorithm cannot achieve. Convincing evidence for this claim is found in more reliable publications that appear in prestigious journals. This algorithm has successfully demonstrated the characteristic ability in identifying preclinical spinal cord injury and traumatic brain injury [58, 90], type-2 diabetes subgroups [57], clinically distinct neurophenotypes in young children [91], a subgroup in asthma and chronic obstructive pulmonary disease [92], relevant genetic alterations in cancer [26], homogeneous subgroups of patients [62], etc.

In addition, the clustering ability of the Mapper algorithm also manifests itself not only for biomedical data but also in many different areas, including social networks [93, 94], epidemiology [95], weather forecasting [96], radar [85, 97], pathology [61], etc.

2.3.2 Feature Selection

Another successful application of the Mapper algorithm is selecting features that best characterize the data. In addition to the global structure shown by the visualization, its obtained result is an information summary at the local level with clusters and their interactions [14]. This visualization allows easy and intuitive discovery of why connections exist within the data. Therefore, when using this algorithm to select features, it should be based on the principle of data shape recognition that encodes the essential structural information [98].

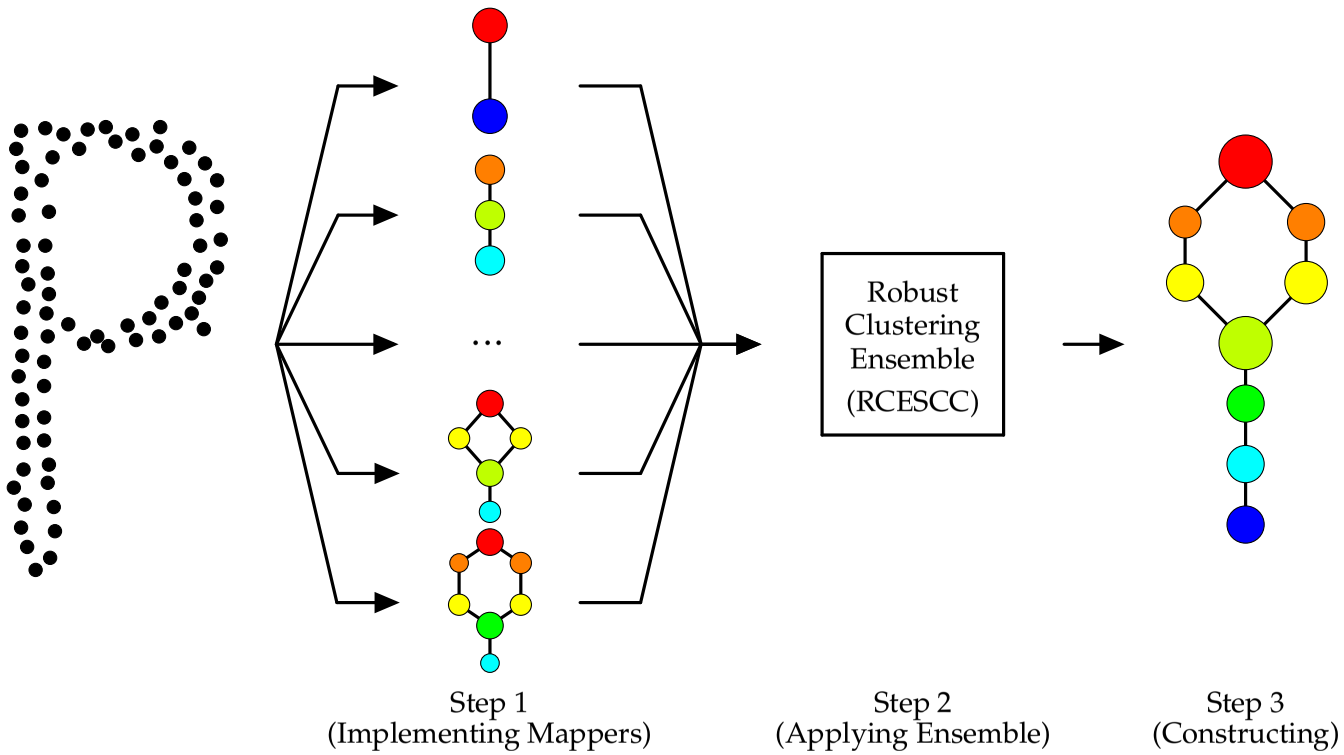


Figure 14: The implementation of the EM algorithm on a point cloud with a noisy P structure.

Unique structures of topological significance often take the form of loops and flares. Those are considered impressions that need attention when observing the intuitive results obtained from the algorithm. The detection of significant subgroups is often carefully evaluated through large changes to the resolution parameters. These subgroups often have an exciting shape that persists steadily despite substantial differences in parameters [98]. Standard statistical techniques are then be used to evaluate and identify features that best differentiate subgroups.

For the Mapper algorithm, feature selection is often associated with data clustering that uncovers meaningful subgroups. This possibility is considered an adjacent step after discovering exciting structures within the data. It helps users gain insight into data that underlies the selection of the essential elements for distinguishing these remarkable structures. Using the obtained result characteristics, including the size and color of the nodes in the graph, also affects the feature selection. Their visualization aids users in decision-making by helping them quickly and intuitively recognize which features are essential to select for evaluation. By evaluating features on Mapper results using foundational statistical techniques, users can find out why interesting data structures are constructed. Therefore, it can be said that feature selection effectively supports explaining the models and making them more meaningful.

The application in feature selection of the Mapper algorithm is as popular as its unique clustering capabilities. This algorithm is quite successful in use to exploring low-density states in biomolecular folding pathways [56], extracting insights from the estrogen receptor gene from breast tumors [17], studying cell reactivity in both type-1 and type-2 diabetes [99], performing a genetic association analysis of the emergent subtypes for type-2 diabetes to identify subtype-specific genetic markers [57], discovering preclinical spinal cord injury and traumatic brain injury [58], predicting manufacturing productivity [98, 100], identifying clinically distinct neurophenotypes in young children with fragile X syndrome [91], identifying traits of hip osteoarthritis [60], recognizing pain using peripheral physiological signals [63], predicting long-term functional outcome in early psychosis [61], identifying relevant genetic alterations in cancer [26], indicating the risk of adverse clinical effects in a cohort of critically ill patients with atrial fibrillation [101], etc.

2.3.3 Data Visualization

Visualization and interaction of the Mapper algorithm are two outstanding characteristics besides the clustering ability and feature selection. When used for clustering data or selecting features, its results are always generated to observe the data shape. However, how the output can be used, as a clustering result or input for machine learning or visualization for observation, all depends on the needs and preferences of the user. In previous applications [56, 89, 17], the visualization of this algorithm was still emphasized in addition to the clustering ability.

With the ability to visualize data for interactive observations, the Mapper algorithm was used to analyze an enormous number of IP packets on network monitoring data to make malicious activities patterns easily observable by security analysts [102]. It can deal with enormous

traffic volumes to extract packet groups belonging to the same activity, especially malicious ones, even if they are all heavily mixed. On the other hand, it is also used to identify and visualize disease microclusters masked by group-level analysis for demonstrating airway pathological heterogeneity in asthma [103]. Further, this algorithm can capture the clustering structure of cells and preserve the continuous gene expression topologies of cells [104]. By combining with gene co-expression network analysis, the differential expression patterns of these modules can be revealed along with the Mapper visualization.

The spotlight for the intuitive and interactive capabilities of the Mapper method is reflected in a famous work [20] on the dynamic organization of the brain. This work focuses on how the brain dynamically adapts for efficient functioning on a functional magnetic resonance imaging dataset. The Mapper algorithm is used to reveal the overall organization of individual whole-brain activity maps as an interactive representation without arbitrarily collapsing data in space or time. This approach distills complex brain dynamics into interactive and behaviorally relevant representations. This work was the main inspiration for the official announcement of the DyNeuSR software package [21] under the BSD License in 2019. This package is much appreciated, especially in its ability to assist users in annotating Mapper-generated networks with metadata, connect them with known neuroanatomical correlations, and study their topologies to capture temporal transitions between coactivated brain patterns. Using the Mapper algorithm in particular and TDA in general for researching neuroscience is one of the great ideas [105, 22] and intensely studied by the Brain Dynamics lab of Stanford University with two principal members Manish Saggar, Caleb Geniesse, and associates [27, 23]. Besides, another research work [106] also provides reliable proof of the effectiveness of the Mapper method in extracting subtle dynamic properties of high temporal resolution magnetoencephalography data without the temporal and spatial collapse. At the same time, it also detects a novel neuroimaging marker, which is hard to see with the other pipelines that require collapsing the data in the spatial and temporal domain.

In addition, the Mapper algorithm also began to have preliminary studies on its applicability in representing 3D volumetric images [107] and 3D printed objects [108]. It is considered an emerging potential tool in the 3D printing area, but further studies are needed to provide specific application directions [109].

2.4 Available Toolkit

The Mapper algorithm was first appeared in 2007 [13] with an original code written by Gurjeet Singh in Matlab. This original code was rewritten by Daniel Müllner on the same platform and published under the GNU General Public License in 2010. This information is obtained from the TDAmapper package copyright provided by Paul Pearson [110].

Since then, many data analysis tools have approached the Mapper algorithm as a stable theoretical framework. One of the famous proprietary tools mentioned is *AyasdiAI*, developed by the Symphony AyasdiAI team [17]. This team, also known as the company, was first founded

by Gurjeet Singh, Gunnar Carlsson, and Harlan Sexton. Their first product, the Ayasdi Iris software, used the Mapper method as the core of its operation. Over time, the AyasdiAI machine intelligence platform uses TDA to create compressed data representations by combining a host of state-of-the-art expertise in machine learning, statistics, and geometry. Its uniqueness is reflected in its good visualization and high interactivity. This capability makes it possible for users to explore the data and clearly understand the meaningful relationships quickly. The development of AyasdiAI is increasingly surpassing expectations as it is being operationalized to address some of the most complex challenges facing modern financial businesses today with a commitment to innovative, comprehensive solutions.

Besides proprietary software, various implementations of this quintessential algorithm are freely available with stable quality and high performance on popular modern programming languages like R and Python. An overview of the public, popular, and free technical toolkits can be found in Table 3, including Python Mapper [111], TDAmapper [110], Kepler Mapper [112, 113, 114], Mapper [115, 116], DyNeuSR [21, 117], Giotto-tda [118, 119], and Mapper Interactive [120, 121]. In addition to these implementations, other packages share the same theoretical framework but are less popular and written specifically for personal purposes. These packages can be found easily on the internet with the source code freely available. To obtain a summary of the commonly available toolkits based on the Mapper algorithm, only those listed in Table 3 are reviewed and evaluated in the following content.

Python Mapper [111] implements the Mapper algorithm written by Daniel Müllner and Aravindakshan Babu in Python. This package was published under the GNU General Public License in 2013 with two options available to the user for this toolchain: a GUI interface and a script package. Each option has its advantages and disadvantages. With a GUI interface, users can conveniently interact to approach the algorithm with a clear layout step by step. The graphical user interface of Python Mapper is not only convenient for non-experts to access, but also speeds up the workflow for beginners and experts alike. However, extending the algorithm with multi-dimensional filter functions has specific difficulties when using the user interface. In contrast, users can easily access the script package by importing the Mapper module and writing their scripts. It is not as convenient as using a GUI interface, but it allows to define filter functions with any dimension. Overall, the authors have created an efficient implementation of the Mapper algorithm that is suitable for many users and published its source code so that anyone with a new idea can customize it easily, smoothly, and quickly.

TDAmapper [110] is programmed in R by Paul Pearson based on cleaning, modifying, and porting the Mapper source code originally written by Daniel Müllner and Gurjeet Singh in Matlab. It also fixes two bugs that existed in the source code. Therefore, this package was also published under the GNU General Public License in 2015 as contributions by Paul Pearson, Daniel Müllner, and Gurjeet Singh. While not a perfect product with an eye-catching interface, the authors of Python Mapper and TDAmapper have always strived to provide a complete, extensible, and fast toolchain to the scientific community. It can be said that these are two foun-

Table 3: The overview of the popular, accessible, and available technical toolkits that use the Mapper algorithm as the core of their operations.

Toolkit	Language	License	Author(s)
Python Mapper [111]	Python	GNU GPL	Daniel Müllner, and Aravindakshan Babu
TDAMapper [110]	R	GNU GPL	Paul Pearson, Daniel Müllner, and Gurjeet Singh
Kepler Mapper [112, 113, 114]	Python	MIT	Hendrik Jacob van Veen, Nathaniel Saul, David Eargle, and Sam W. Mangham
Mapper [115, 116]	R	MIT	Matthew Piekenbrock
DyNeuSR [21, 117]	Python	BSD	Caleb Geniesse, Olaf Sporns, Giovanni Petri, and Manish Saggar
Giotto-tda [118, 119]	Python	GNU AGPL	Guillaume Tauzin, Umberto Lupo, Lewis Tunstall, Julian Burella Pérez, Matteo Caorsi, Wojciech Reise, Anibal Medina-Mardones, Alberto Dassatti, and Kathryn Hess
Mapper Interactive [120, 121]	Python	MIT	Youjia Zhou, Nithin Chalapathi, Archit Rathore, Yaodong Zhao, and Bei Wang

dational packages written in two popular programming languages, Python and R, which are of great significance to scientists studying the Mapper algorithm. They can be considered the first bricks that laid the foundation for creating free implementations of the Mapper algorithm for the community. Moreover, they are the main driving force behind new packages becoming more and more perfect in user interface and data visualization. In 2020, TDAmapper is used as an online tool for TDA and visualization, *TDAview* [122]. It is a user-friendly tool that allows biologists and clinicians with no programming knowledge to harness the power of TDA. This package is published open-source under the MIT License by Kieran Walsh and Kamile Taouk [123]. It can handle massive datasets, so it has many applications for high-dimensional data, including the construction of topology-based gene co-expression networks.

Kepler Mapper [113] is a flexible Python implementation of the Mapper algorithm. It has been around since 2017 [112] with two principal authors, Nathaniel Saul and Hendrik Jacob van Veen. This package has evolved more and more entirely and was distributed officially in 2019 under the MIT License in collaboration with four authors, Hendrik Jacob van Veen, Nathaniel Saul, David Eargle, and Sam W. Mangham [113]. Kepler Mapper brings Mapper algorithm to users with an intuitive interface and provides them with various comprehension methods in visualizing the network graph generated by this algorithm [114]. It gives the ability to analyze Mapper network graphs intuitively with many techniques, including interactive visualization and exploration in the browser, visualization interface for embedded purposes, and static visualizations. Its flexibility and user-friendliness come from leveraging Scikit-Learn-API-compatible [124, 125] scaling and clustering algorithms in constructing network graphs [113]. Although it is not a professional or mature product as AyasdiAI, Kepler Mapper tries to provide the scientific community with a user-friendly, fast, and extensible package with an open-source. This implementation is well-recognized in the community as it is widely used as a framework for scientific experimentation or new version development. The suite of tutorials on using Kepler Mapper for simple and complex use cases is publicly and widely available on its homepage. This package was recently developed and improved as a library in the *Scikit-TDA* project [126, 127] that provides TDA Python tools in widely usable and easily approachable forms.

Mapper [115] was introduced as a theoretical framework-based implementation of the Mapper algorithm in R. It was written by Matthew Piekenbrock and distributed under the MIT License in 2019 [116]. This package is constructed to effectively implement the Mapper method through practical default parameter settings for those unfamiliar with this algorithm in an excellent intuitive and interactive way. It is also designed to modify or extend the core framework more simply, smoothly, quickly, and efficiently, without limiting its generality [115]. All information about this current development release can be found on its homepage [116].

DyNeuSR [21] is a Python visualization library for exploring, analyzing, and validating topological representations of neuroimaging data. It is specially designed to work with network graphs extracted by the famous TDA method, the Mapper algorithm. This library is the

realization of a good idea that uses TDA to reveal the overall organization of individual whole-brain activity maps without arbitrarily collapsing data in space or time [20]. DyNeuSR was published under a BSD License in 2019 by the Brain Dynamics lab of Stanford University with two principal members Manish Saggar, Caleb Geniesse, and associates [117]. Kepler Mapper [126] is used as its core to implement the Mapper algorithm to visualize graphs interactively and analyze neuroscience. The highlights of DyNeuSR are immensely appreciated, especially in the ability to assist users in annotating Mapper-generated networks with meta-information, connecting them to known neuroanatomical correlates, and studying their topological characteristics to capture temporal transitions between coactivated brain patterns [21]. It is also an open-source project, so the code, documentation, and example tutorials are publicly and widely available on the package's homepage. The authors hope that DyNeuSR can enable and encourage the larger neuroscience community to more effectively mine Mapper-based methods to generate biological insights underlying complex mental disorders.

Giotto-tda [118] is a high-performance topological machine learning toolbox written in Python based on the Scikit-learn library [124] and distributed under the GNU Affero General Public License in 2021 [119]. It is a member of the Giotto family of open-source projects. Giotto-tda inherits the flexibility of the famous library [124] and retrofits topological analysis capabilities based on the popular major theoretical frameworks of TDA, including persistent homology and Mapper algorithm. It allows TDA to apply to univariate and multivariate time-series, images, graphs, and even geometrical structures with high dimensionality and simple complexes. This ability makes Giotto-tda the most comprehensive Python library for topological machine learning and data exploration to date. In terms of implementation for the Mapper method, its design provides a great deal of intuitive interoperability and computational efficiency. It also allows users to realize each respective operational stage in the algorithm, effectively integrate Mapper pipelines as part of a larger machine learning workflow, and optimally use memory caching to avoid unnecessary news in re-computations [118]. These are considered the highlights of Giotto-tda when compared to Keppler Mapper [126]. To the best of the authors' knowledge, Keppler Mapper does not deploy all algorithm steps in a single class and is only partially compatible with the Scikit-learn library [124, 125]. Furthermore, it does not support memory caching or provide real-time hyperparameter interactivity in the visualization. Giotto-tda also provides the code, documentation, tutorials, and illustrations publicly and freely according to the open-source statement on its homepage [119].

Mapper Interactive [120] was introduced in 2021 as a web-based platform for interactively analyzing and visualizing a high-dimensional point cloud. This package is distributed officially under the MIT License in Python [121], in which its operational core is the Mapper algorithm. The contribution of this implementation is highlighted in three aspects: extendability, interactivity, and scalability to support real-world data analysis. Moreover, its optimization, efficiency, and speed are also demonstrated in comparative correlation with the two most modern applications, Kepler Mapper [126] and Giotto-tda [118]. Mapper Interactive is emphasized on

its ability to make the Mapper method accessible to non-specialists and speed up topological analytics workflows.

In summary, although freely available with open source, the packages that approach the Mapper algorithm as a solid theoretical framework are increasingly mature and stable. They aim to provide a complete, user-friendly, quick, scalable, and efficient toolchain to the scientific community. Almost all packages publish their code, documentation, tutorials, and illustrations publicly and freely on their homepage. Because each tool has different strengths and weaknesses, the choice of which tool for implementing this algorithm depends on each researcher's actual needs, personal knowledge, and personal fortes.

2.5 Current Limitation

Besides the advantages in analyzing, visualizing, and understanding data, the Mapper algorithm still has certain limitations when implemented in science and practice. The simplicity of the Mapper design leads to an abundance in the choice of user parameters. How to choose the parameters to obtain the desired results becomes a challenge for Mapper users. In addition, the obtained results after implementing this algorithm are not stable and very sensitive to these parameter choices. Therefore, evaluating the efficiency of Mapper-type algorithms, which both cluster data and detect data structure, becomes a problem that requires users to pay attention when using.

The choice of filter function depends on the features of a particular data set. The obtained output is susceptible to this choice, so it is important. There is no recommended rule to choose in practice [36]. The filters that are often used popularly are oriented from previous famous and successful works. However, these orientations depend significantly on the characteristics of each data set, and there is no advice for any data set. The idea for optimizing the filter choice is also interesting, but this is a big challenge. Besides, another idea has also been considered to remove Mapper's dependency on the filter by not using it when structuring the algorithm. This idea is good, but it should be noted that the filter is considered a Mapper's characteristic when using the pre-image of a continuous function to construct the simplicial complex on point clouds instead of using other simplicial complexes Čech complex, Vietoris-Rips complex [128], witness complex [129], geometric complex [130], etc. In general, the filter choice in the Mapper algorithm is still a challenge and can be viewed as a data-driven decision-making problem.

The filter range is covered by regular intervals such that two adjacent intervals overlap by the same percentage. The number of intervals and the overlapping percentage are known as the Mapper algorithm's resolution parameters. The resolution meaning is expressed by correlating these parameters for features of the obtained graph after implementation. While the intervals reflect the nodes, the overlapping percentage demonstrates the connectivity between the nodes. The output graph created by many intervals has more nodes. A more significant overlapping percentage also means more edges, i.e., more connections, in the graph. The dataset size is one of the essential criteria for changing the resolution parameters. It is easy to see that larger

datasets need more intervals to represent data complexity, and the overlapping percentage can be lowered to control the number of edges in the data representation effectively. A tiny change to the resolution parameters can substantially change the Mapper output, so the algorithm results are often unstable. Usually, these parameters are fine-tuned until the obtained results capture some exciting features of the dataset and satisfy user exigency. Then, the cover corresponding to these resolution parameters is well-chosen [36].

The same distribution for the cover intervals on filter range is also unnatural for the Mapper algorithm. It makes more sense to cover this range if it is generated data-driven. When using this algorithm with many filter functions, covering the filter range is also a challenge for users. In addition, when using this algorithm with many filter functions, the cover formation is an issue that needs to care. Almost current works use the rectangular cover for the Mapper algorithm with two filter functions. However, at least two cover methods exist for the two-filters case mentioned in the original work. Therefore, covering the filter range in the case of many functions is also a particular challenge. Furthermore, how the clustering algorithm is chosen to achieve the desired result is also very diverse for Mapper users.

The Mapper evaluation has traditionally been based on two criteria: clustering and visualization. As for the clustering ability, it is usually evaluated by traditional metrics. As for the visualizing ability, it is often assessed with the naked eye with the visual color painted at the user's discretion to highlight the critical feature(s) in the obtained result. However, there needs to be a technique that evaluates the Mapper algorithm based on its unique abilities. This research direction is still in its infancy, with work on the numerical measure for the instability of Mapper-type algorithms [131]. This method measures the variability of the obtained outputs as a function of the parameter choice. The authors derive theoretical results that propose estimates for the instability and provide practical ways to control this instability. These results are just the beginnings of suggesting other tools evaluating the Mapper algorithm as a unique data analysis method. Moreover, these new tools are also going to open promising directions for more and more complete Mapper optimization.

2.6 Future Direction

The limitations of the Mapper algorithm at present serve as the impetus for its development and evolution in the future. These motivations will help the community have research orientations that are close to the scientific and social trends in the coming time. These orientations also revolve around two main aspects, including theory and application.

The Mapper algorithm is constructed from the inspiration of the Reeb graph and has a close relationship with discrete Morse theory [13]. Theoretical studies as its working foundations are more and more complete and developed. They often focus on making a solid fulcrum from classical mathematical theories so that this algorithm can work optimally, including stability [132], statistical selection for parameters [133, 35, 134], performance evaluation [135, 131, 136], foundation for machine learning and deep learning [137, 138, 70, 139], forming the next gener-

ation [115, 140, 141], etc. It soon develops rapidly with many publications as the connection of the Mapper algorithm with the Reeb graph and Morse theory is more and more intimate [142]. This orientation requires the scientists to understand of both mathematics, computer science, and other related fields. In addition, it continues to evolve simultaneously using this method to solve applied problems with real-life data so that the applications are always reliable for users [14].

The most potent application area of the Mapper algorithm in the next few years could be in biomedicine, as evidenced by the explosive beginnings of peer-reviewed publications in prestigious journals [91, 25, 20, 26, 14, 101, 143]. Its effectiveness is increasingly proven through clustering with its ability to detect significant subgroups [92, 144, 145, 62], select useful features to process [63, 29, 62], and visualize with friendly interactions [21, 113, 118, 120] for biomedical data. Two promising lands that are being vigorously explored by TDA tools today are the brain with neuroscience [20, 23, 146] and gene structure in pathology [106, 147, 148, 149]. It also shows its adaptation to hot current world problems such as the Covid-19 pandemic [150, 151, 55, 152]. The success of this algorithm, as well as the TDA method in supporting the diagnosis and treatment of diseases, comes from the unique characteristics of the computational topology. In the future, this orientation will develop even more intensely on the big bricks that have been firmly laid by previous famous publications [136, 28].

Besides, the application of Mapper algorithm is also shown in its adaptability in other related fields including biology [153], physics [154], chemistry [155, 156], environment [157, 158], industry [159], 3D printing [107, 109], remote sensing [85, 97], economy [77, 160], medicine [19], earth science [161, 162], etc. The prospect of using it in the direction of applications in other fields is very high, beneficial for structured data types with high dimensionality. This orientation is also one of the potential new ideas to spread the Mapper algorithm effectively.

Today, artificial intelligence, which has grown enormously and rapidly thanks to machine learning and deep learning, presents significant opportunities and challenges for data analytics tools. The Mapper algorithm has also demonstrated its potential in this area because of its unique capabilities. It is becoming an effective method to bring high performance to machine learning and deep learning models [163]. They are often used as clustering or feature selection methods when combined with other techniques that are effective for data analysis [101, 164, 158, 155]. In addition, several theoretical foundations using them as the core are also formed as a new promising model of machine learning [137, 138]. The Mapper algorithm carefully enters deep understanding because of its solid theoretical frameworks [28, 165, 139]. It is like new energy that makes deep learning more powerful [166, 167, 163, 70, 168]. In addition, one of the prospects can soon be mentioned Mapper's interpretability to learning models to approach explainable artificial intelligence [169, 170, 171].

One of the future expectations of the Mapper algorithm is its ability to handle time-series data [172, 173]. It is also a good land for groundbreaking announcements [174, 160, 175, 176, 177]. This research direction has been partly reflected in the brain representation by the visual-

ization capabilities of this algorithm [20, 21]. Furthermore, in the last few years, this algorithm has emerged with the ability to provide a robust theoretical framework for summarizing network data while preserving their core structures [67]. Using this algorithm to reconstruct the original graph more compactly while maintaining its structure is a novel and possible idea. This idea can be considered a pre-processing step before mining large or giant graphs [70]. This orientation is promising in graph approximation mining because of its adaptability to high-dimensional network data. However, these ideas need a specific time to be perfected in theory and application before practical implementation.

2.7 Discussion

This chapter has systematized the understanding of the Mapper algorithm from theory to the application as an excellent representative of TDA. This algorithm has been carefully restated with deep insights and intuitive illustrations. Its variations are also systematically aggregated to clarify its evolution over time. The effectiveness of the Mapper algorithm in data analysis is presented clearly and logically through its major popular applications, including clustering, feature selection, visualization, and support for artificial intelligence. In addition, this work also provides a concise introduction to the popular out-of-the-box tools that work on the Mapper frameworks at its core. Moreover, its current limitations are also recognized as a fulcrum for future orientation in studying and developing this algorithm.

Although this work has been carefully elaborated, it is undoubtedly not without shortcomings. This limitation may come from the massive explosion in research on the Mapper algorithm in recent years, both in theory and in applications. Theoretical studies are increasingly ensuring certainty for applying it in real-life data. Generally, this algorithm is considered a promising candidate for TDA with powerful and practical capabilities.

3 Fuzzy Mapper Algorithm

Chapter 3 proposes a new method of improving the Mapper algorithm, called *Fuzzy Mapper*, by using overlapping clustering for covering the filter range naturally. In this method, the FCM algorithm, a famous representative of overlapping clustering, is used as the primary tool to optimize the choice of cover. The FM algorithm is thus introduced as a development of the Mapper algorithm in dividing cover intervals automatically with an arbitrary overlapping percentage. Its advantages are also shown through extracting insights and meaningful information from high-dimensional datasets in many fields, especially bioinformatics and neuroscience. Three real-world datasets, including Unit Circle, Reaven and Miller Diabetes, and NKI Breast Cancer, are implemented to clarify this algorithm's effectiveness. The results are analyzed in the comparative relationship between two algorithms, FM and Mapper, through the output shape in the topological sense and silhouette coefficient score in clustering evaluation.

Overall, the contributions of Chapter 3 can be summarized as follows:

[C3.1] Proposing the FM algorithm by optimizing the cover choice to automatically divide the filter range into irregular intervals with random overlapping percentages.

[C3.2] Demonstrating that the FM algorithm can generate outputs similar to the Mapper algorithm from a topological standpoint.

[C3.3] Reporting well-clustered results based on the FM algorithm's silhouette coefficient score in most experimental cases compared to the Mapper algorithm.

Moreover, the rest of this chapter is organized as follows. Section 3.1 presents the motivation for proposing the FM algorithm. Section 3.2 describes the algorithm carefully with visual illustrations. The experimental results on three real-world datasets are reported clearly in Section 3.3 based on the comparison between this proposed algorithm and the original algorithm. Finally, the conclusions in Section 3.4 review what has been achieved to orientate and develop future researches on the FM algorithm.

3.1 Motivation

In the Mapper algorithm, the length of small intervals and the percentage of overlap between consecutive intervals is called the resolution parameters. The length of intervals can be used equivalently by the number of intervals because of the negative correlation. When the overlapping percentage is fixed, the longer the length of small intervals is, the smaller the number of intervals is, and vice versa. Moreover, the number of intervals is positively correlated with the number of nodes in the output shape. When the number of intervals decreases (increases), the number of nodes also decreases (increases) correspondingly. Similarly, the overlapping percentage is also positively correlated with the connectivity in the output

shape. When the overlapping percentage increases (decreases), the degree of connectivity is higher (lower).

The remarkable thing is the number of intervals and percentage of overlap is extremely sensitive to the output shape. It is often unstable because a slight change in resolution parameters can lead to a considerable difference in the output. In Figure 15, the results of the Mapper algorithm from these two situations are very different, although using the same dataset. The cover is well-chosen if the output is deemed the most informative form from the user's perspective. Moreover, covering the filter range by regular intervals and same overlapping percentage is one weak point in the choice of cover. This choice inadvertently causes unnatural phenomena in the covering method, and it can be improved by using a clustering process.

Overlapping clustering is more natural and effective than exclusive clustering in many real situations. This clustering randomly separates data points into clusters with different overlapping percentages. It is quite suitable and compatible with the cover choice step for the filter range in the Mapper algorithm. According to the topological meaning, these clusters play a role as a cover of the filter range. So, there are three leading causes why overlapping clustering is a bright and promising candidate to perfect the Mapper algorithm in choosing cover more naturally:

(1) Firstly, it is more flexible since each object can belong to two or more clusters.

(2) Secondly, the overlapping percentage between clusters is unforeseen and random.

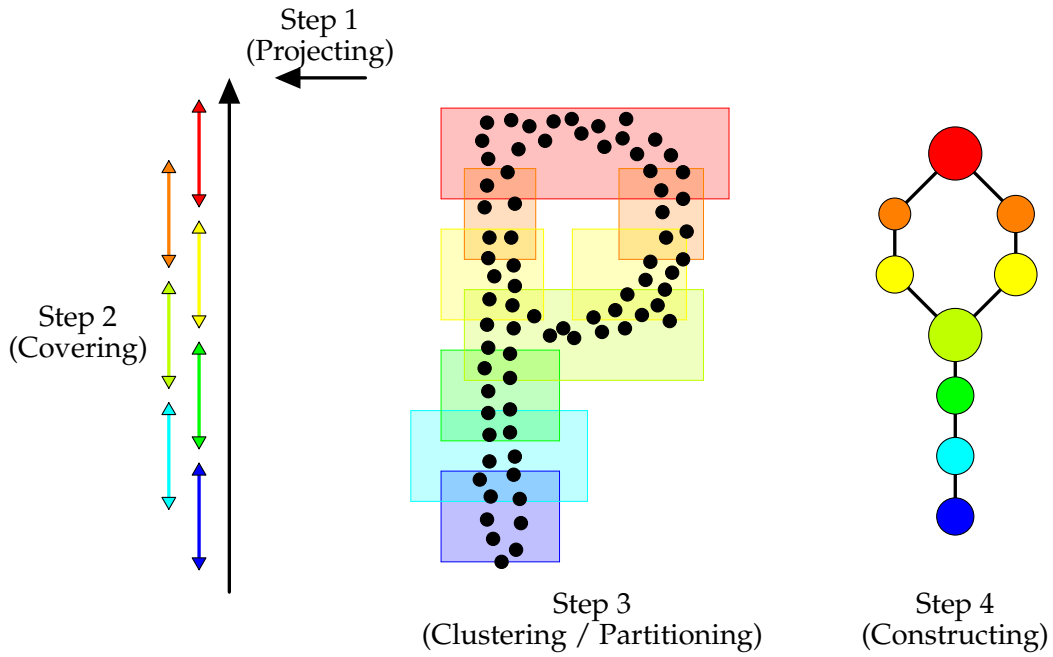
(3) Thirdly, clusters are separated reasonably based on the density of data points.

In conclusion, the Mapper algorithm needs to optimize the choice of cover on the filter range. One of the brightest candidates to meet this demand is overlapping clustering. Therefore, the FCM algorithm, the most famous representative of overlapping clustering, is used to develop the Mapper algorithm in this situation. How does it perfect the original algorithm to construct a novel algorithm? What are the similar and different points of two original and improved versions? These questions are discussed in the next section.

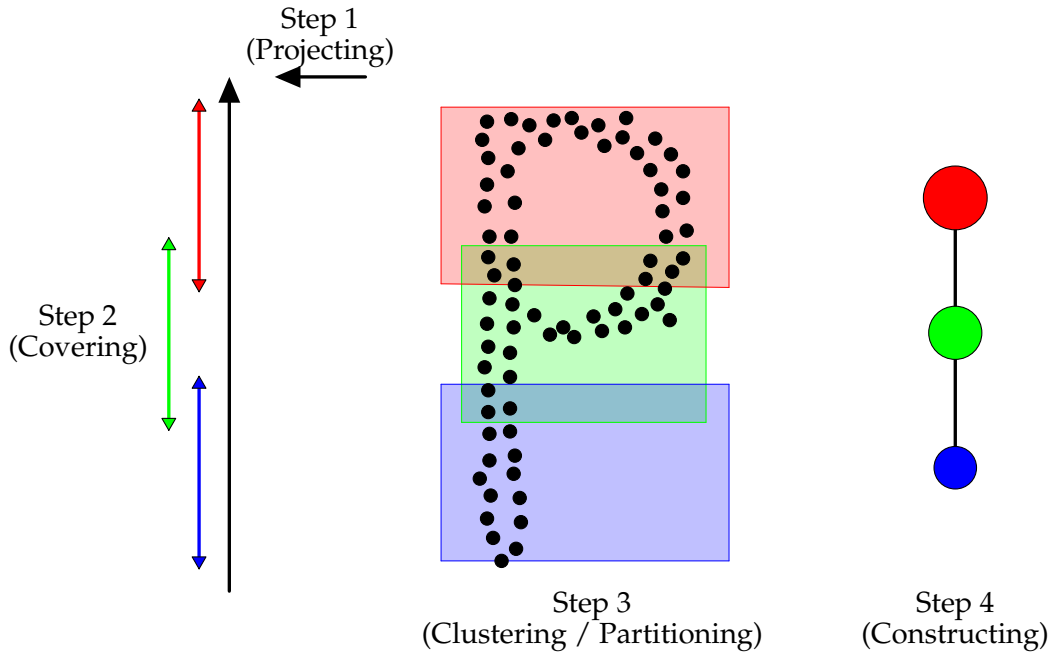
3.2 Description

In the same way, as with the Mapper algorithm, the FM algorithm uses a discrete dataset $\mathbb{X}$ as its *input*. This algorithm starts working with four user-defined *parameters* as follows:

- Distance metric d is used to calculate distances between data points for clustering or partitioning.
- Filter function $f : \mathbb{X} \to \mathbb{R}$ projects all data points in $\mathbb{X}$ to $\mathbb{R}$.
- Cover $\mathcal{I}$ of the filter range $f(\mathbb{X})$ with is responsible for dividing the filter range automatically into N irregular intervals and the random overlapping percentage between them.



(a) 7 intervals and 40% overlap



(b) 3 intervals and 20% overlap

Figure 15: The illustration of the Mapper algorithm on a sampled point cloud with a noisy P structure for different resolution parameters.

Table 4: The description of the FM algorithm.

Input	Dataset $\mathbb{X}$ with finite elements.
Parameters	– Distance metric d, – Filter function $f : \mathbb{X} \longrightarrow \mathbb{R}$, – Cover $\mathcal{I}$ of the filter range $f(\mathbb{X})$ with N irregular intervals and the overlapping threshold τ. – Clustering algorithm $\mathcal{C}$ (option).
Method	1: Projecting all data points in $\mathbb{X}$ to $\mathbb{R}$ by using the filter f. 2: Covering $f(\mathbb{X})$ by N irregular intervals and the random overlapping percentage between them based on the threshold τ. 3: Decomposing the pre-image $f^{-1}(I_i)$ of each cover interval I_i into clusters $C_{i,1}$, $C_{i,2}$, $C_{i,3}$, ..., C_{i,n_i} by using the clustering algorithm $\mathcal{C}$ or partitioning connected components based on the distance metric d. 4: Constructing the simplicial complex $\mathcal{G}$ defined by clusters C_ij and their intersections.
Output	Simplicial complex $\mathcal{G}$ as a geometric representation of $\mathbb{X}$.

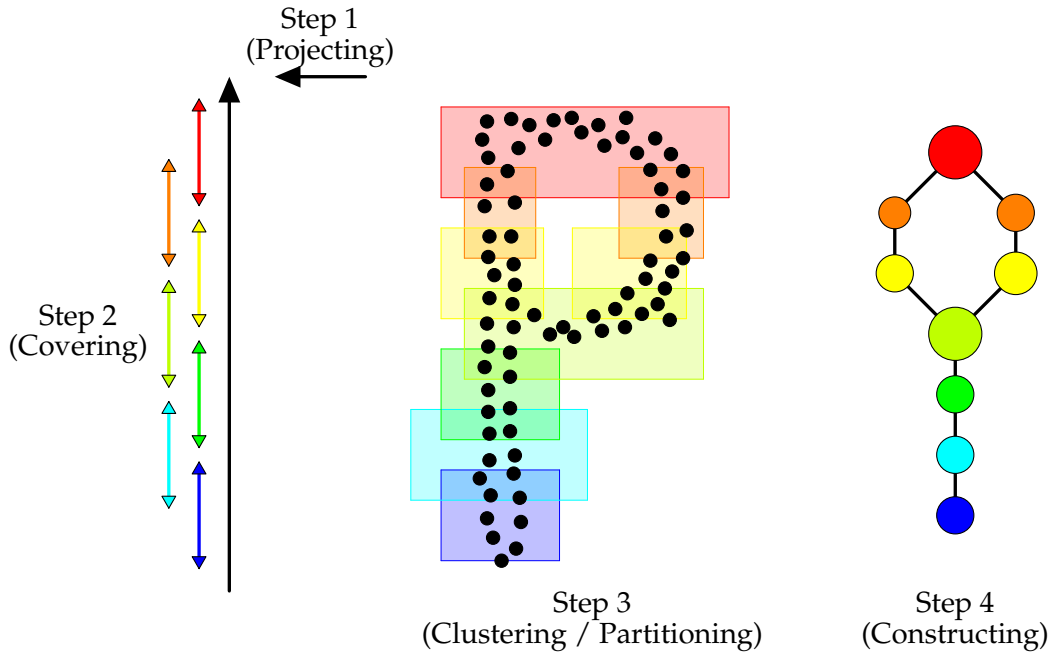
This overlap determines the connectivity between two intervals and depends on the threshold τ. The output of the FCM algorithm is seen as the reasonable cover $\mathcal{I}$ on the filter range $f(\mathbb{X})$.

– The clustering algorithm $\mathcal{C}$ is used to separate data points in every element of the pull-back cover induced by the pair $(f,\mathcal{U})$. This algorithm changes the representation from a topological version to a statistical one. As well as the Mapper algorithm, the Mapper algorithm does not depend on a specific clustering algorithm. So, in addition to using clustering algorithms, users can replace this process by partitioning connected components.

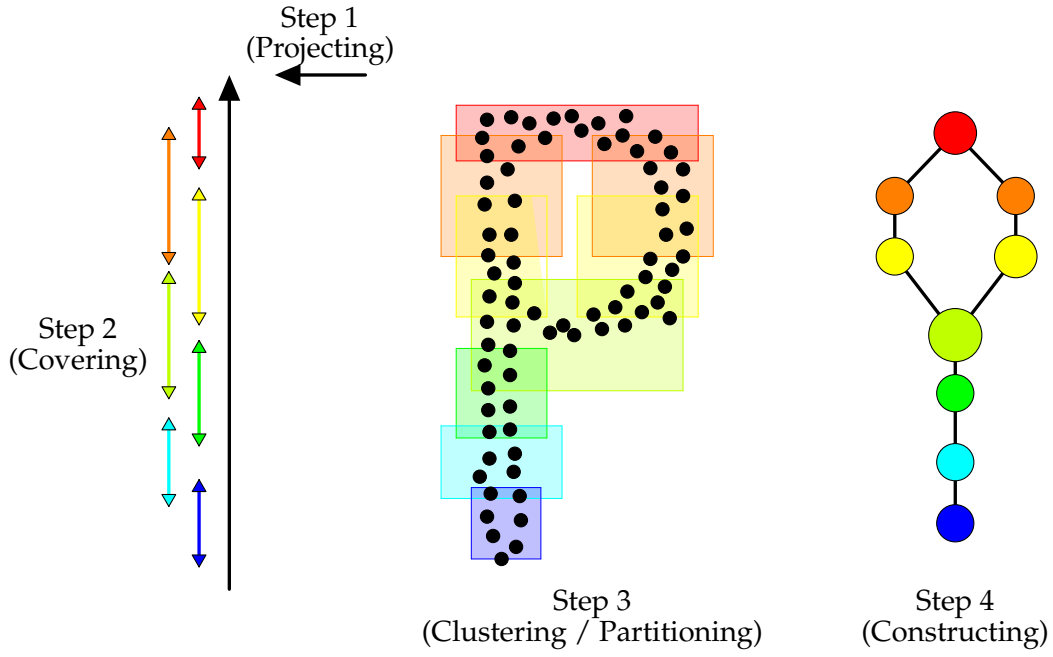
This novel algorithm's *output* is also a network graph or a simplicial complex, in general, that is a *geometric representation* of the dataset $\mathbb{X}$ that consists of nodes and edges. The color and the size are the two main characteristics of nodes. The color usually indicates the average of the colored values of points, while the size usually instructs the number of points. The blue and red sequentially display the minimal and maximal values, respectively. The colors ranging from blue to red express the colored values ranging from low to high.

The FM algorithm is described in detail with the steps in Table 4. It is compared with the original algorithm, specifically in Table 5. Moreover, both algorithms are also illustrated in Figure 16 on the same point cloud sampled with the noisy P structure.

The cover on the filter range in each algorithm depends on two quantities known as the resolution parameters. In the Mapper algorithm, these parameters consist of the number N of regular intervals and the same overlapping percentage p. The FM algorithm consists of the number N of irregular intervals and the overlapping threshold τ that creates a random overlap between intervals. Using a threshold for some limited purposes is one of the popular techniques in data mining, especially fuzzy theory [178, 179]. In this situation, the threshold τ



(a) Mapper algorithm



(b) FM algorithm

Figure 16: The illustration of algorithms on a sampled point cloud with a noisy P structure.

Table 5: The comparison between algorithms in the corresponding steps.

Step	Mapper algorithm	FM algorithm
1	Projecting all data points in $\mathbb{X}$ to $\mathbb{R}$ by using the filter f.	
2	Covering the filter range $f(\mathbb{X})$ by a cover $\mathcal{I}$ that depends on the following resolution parameters: – N regular intervals, – Same overlapping percentage p.	Covering the filter range by a cover $\mathcal{I}$ that is obtained by the FCM algorithm and depends on the following resolution parameters: – N irregular intervals, – Random overlapping percentage based on the threshold τ.
3	Decomposing the pre-image $f^{-1}(I_i)$ of each cover interval I_i into clusters $C_{i,1}$, $C_{i,2}$, $C_{i,3}$, …, C_{i,n_i} by using the clustering algorithm $\mathcal{C}$ or partitioning connected components based on the distance metric d.	
4	Constructing the simplicial complex $\mathcal{G}$ defined by clusters C_ij and their intersections.	

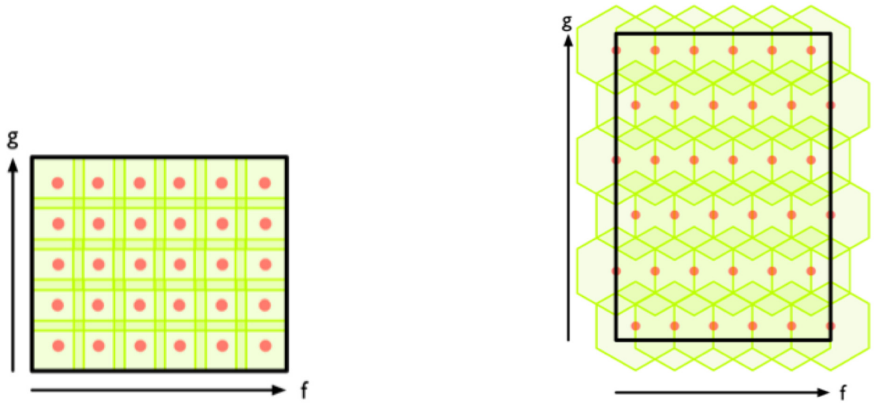
decides the overlapping percentage between intervals by comparing the membership degrees to the overlapping threshold. A data point belongs to an interval if its membership degree is over the threshold τ. Therefore, it can belong to one or more intervals. Moreover, the overlap is created with random and unforeseen percentages, depending on the distribution of data points. The resolution parameters improved by the FCM algorithm are a highlight to make it more natural and reasonable in choosing cover on the filtering range.

From the description and implementation, the FM algorithm outperforms the Mapper algorithm on the following aspects:

(1) Firstly, the overlap in the Mapper algorithm only occurs in pairs and for adjacent intervals of the filter range, but this does not happen in the FM algorithm. One thing to remember for this algorithm is that the overlap between intervals cannot be necessarily consecutive and pairwise.

(2) Secondly, it is straightforward to see that if only using one real-valued function in the original method, the output is a simplicial complex with a dimension no greater than 1. So, to create a multi-dimensional simplicial complex, users need to use it with more than one real-valued filter or a vector-valued filter. However, the output in the novel method can be a simplicial complex with the dimension greater than 1, although only using one real-valued filter. It depends on the overlapping threshold used in the FCM algorithm. This substantial property of the FM algorithm makes it valuable and influential in extracting the high-dimensional topological data structure.

(3) Thirdly, the choice of cover in the Mapper algorithm is very complicated in the case of more than one real-valued filter. There is no standard rule yet for this choice. When the



(a) Rectangles **(b)** Hexagons

Figure 17: Two covering methods are illustrated in the original paper for the Mapper algorithm with two real-valued filters.

Mapper algorithm was first introduced [13], two covering ways were illustrated, including rectangles and hexagons, see Figure 17. It is easy to see that the dimension of all simplexes in the created simplicial complex is insufficient from 0 to the number of filters. Some topological structures are missed if the choice of cover is not good. So, the option is difficult for the Mapper algorithm with multiple filters. The FM algorithm can surmount this because of its flexibility and automation in covering the filter range through the overlapping threshold.

(4) Finally, the cover generated in the FM algorithm is more reasonable and suitable because the FCM algorithm creates the intervals based on the density and distribution of data points.

This chapter uses the FM algorithm in the abridged form instead of the general form. The abridged form is explained in that the relationship between intervals is only considered in a pairwise manner, although the number of intervals that intersect can be over 2. Therefore, the output is a simplicial complex with a dimension no greater than 1. In other words, the FM algorithm in abridged form creates a graph similar to the Mapper algorithm in a simple form with a real-valued function. This abridged form can be understood as a plane shape of the general form. For a careful understanding of the distinction between these forms, an illustration is clearly shown in Figure 18. The mentioned FM algorithm is always understood in the abridged form unless otherwise specified.

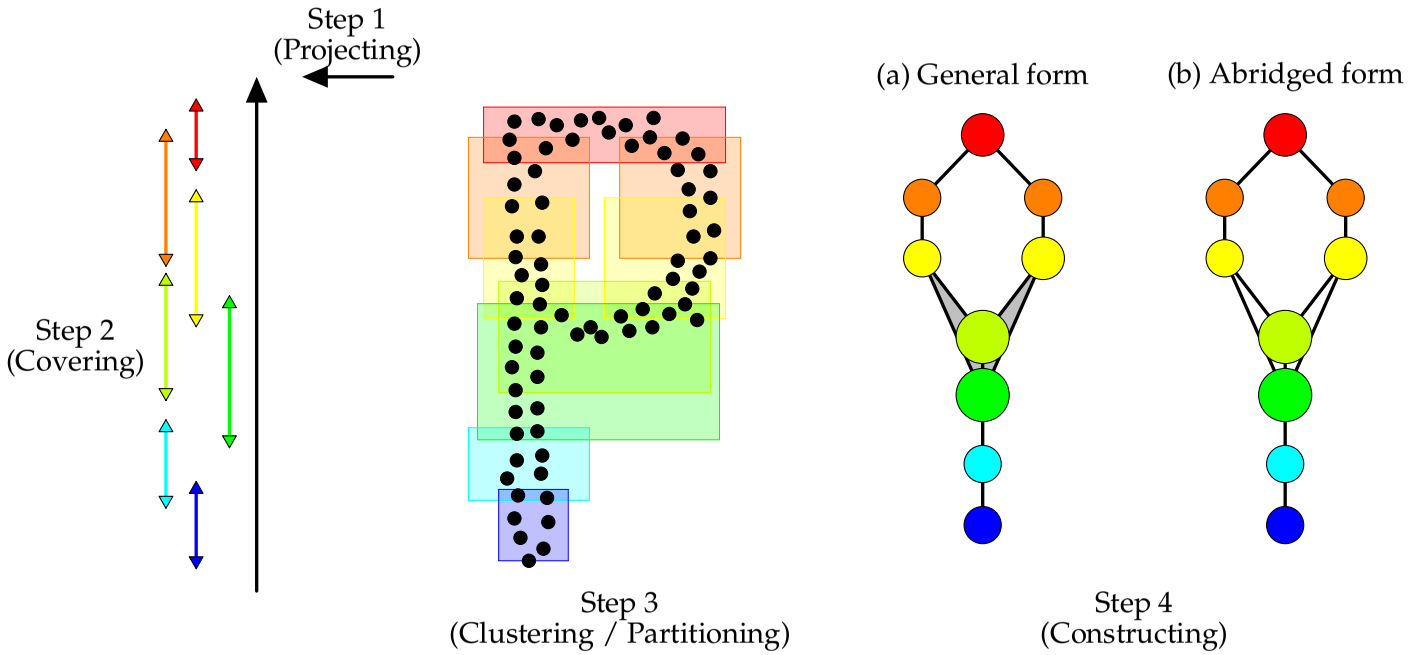


Figure 18: The illustration for two forms of the FM algorithm.

The FM algorithm is only different from the original one in covering the filter range. It is developed naturally by dividing the filter range into irregular intervals automatically with random overlapping percentages. Considerable attention for both algorithms is whether their output is similar or not. By visualizing the numerical datasets, their outcomes are relatively similar in a topological sense. The similarity in the visualization of a point cloud sampled with the P structure by both algorithms is shown in Figure 16. It should be noted that this is only a subjective conjecture based on the shape of some artificial datasets. The efficiency of the FM algorithm needs to be validated by testing with real datasets.

The following section discusses the datasets' characteristics and experimental evaluations to compare the results between Mapper and FM algorithms.

3.3 Experiment

There are many software packages that employ approaches based on the Mapper algorithm. This implementation uses the KeplerMapper, a free Python package for the Mapper algorithm by Saul and Veen [112, 113], to conduct the experiments. Although it is not a professional or mature product, the authors try to provide the scientific community with a complete, user-friendly, fast, and extensible tool. Since its source code is open, it is extended and modified to create source code for the FM algorithm.

In the previous well-known publications, the Mapper algorithm has proven to have several advantages compared to some classical clustering techniques, such as single-linkage, K-means, and PCA. Some papers focus on reaching their clustering performances to highlight the effectiveness of this algorithm in identifying various meaning sub-groups clearly and uniquely [56, 89, 17, 93, 94]. So, in this section, we only examine the efficiency of the FM algorithm by evaluating it with the Mapper algorithm to accentuate this novel algorithm. This evaluation is conducted by comparing the output shape in a topological sense and the internal assessment of the clustering process. Therefore, our experiments are implemented sequentially on three datasets with detailed descriptions.

Each dataset is alternately mined using both two algorithms, Mapper and FM. For each dataset, the parameters concerning the filter, number of intervals, and clustering algorithm are kept the same for both algorithms. The overlapping percentage p in the Mapper algorithm is selected in the same way as in previous well-known experiments for each dataset [13, 17]. The overlapping threshold τ in the FM algorithm is chosen to prove that it can create the output shape somewhat similar to the Mapper algorithm from a topological standpoint.

Moreover, for each dataset, the clustering results of both algorithms, Mapper and FM, are evaluated based on the internal indices. Three internal indices are considered rather suitable candidates, including the Davies-Bouldin index, the Dunn index, and the silhouette coefficient. Both algorithms overlap in clustering, but the enhanced algorithm usually generates more total samples than the original algorithm. Therefore, the Davies-Bouldin and Dunn indices are not chosen as they are sensitive to large total samples and overlapping clustering. The sil-

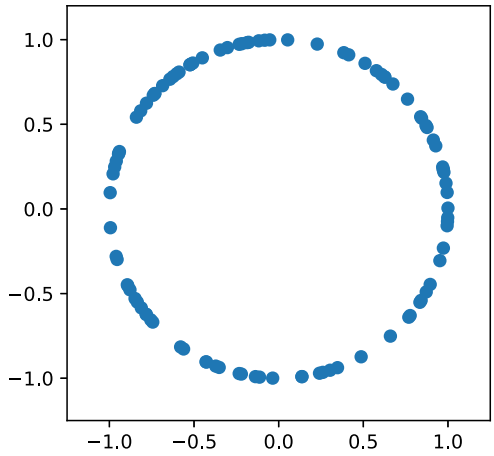


Figure 19: The visualization of the Unit Circle dataset.

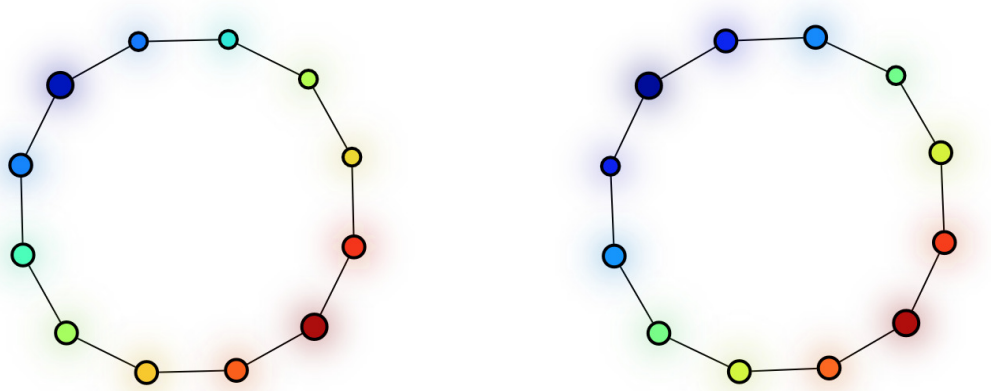
Table 6: The parameter settings for the Unit Circle dataset.

Mapper	FILTER FUNCTION	NUMBER OF INTERVALS	OVERLAPPING PERCENTAGE	CLUSTERING METHOD
	Sum	$N = 7$	$p = 50\%$	DBSCAN
FM	FILTER FUNCTION	NUMBER OF INTERVALS	OVERLAPPING THRESHOLD	CLUSTERING METHOD
	Sum	$N = 7$	$\tau = 0.1$	DBSCAN

houette coefficient is chosen because of its suitability in measuring the cohesion of an object to clusters. The silhouette coefficient is a method to measure the overall quality of clustering [180]. It compares the average distance to elements in the same cluster with the average distance to elements in other clusters. The silhouette value is in the closed interval $[-1; 1]$. The values near zero indicate overlapping clusters, while a negative value generally shows that the sample corresponding to that value has been assigned to the wrong cluster. Samples with a high silhouette value are considered well clustered; otherwise, they may be outliers [180]. The silhouette coefficient score is the mean of the silhouette coefficients of all samples.

3.3.1 Unit Circle Dataset

The Unit Circle dataset has approximately 100 points on the circle with unit radius, centered at the origin in the Cartesian coordinate system in the Euclidean plane. This dataset is widely used as a basic illustration for the Mapper algorithm [13, 93, 94]. The visualization of the Unit Circle dataset is presented in Figure 19.



(a) Mapper **(b)** FM

Figure 20: The outputs for the Unit Circle dataset.

Table 7: The experimental results for the Unit Circle dataset.

	NODES	EDGES	UNIQUE SAMPLES	TOTAL SAMPLES
Mapper	12	12	100	131
FM	12	12	100	125

Both algorithms were used to analyze the Unit Circle dataset in this experiment. Their corresponding parameters are described in Table 6. The Euclidean distance is considered a metric for this dataset. The filter is the sum function of component coordinates. The clustering algorithm is DBSCAN that uses default parameters from the scikit-learn library. The node's color indicates the average lens values at points in the set represented by the node. The red expresses a high value, and the blue defines a low value. The node's size indicates the number of points in the set represented by the node. This size positively correlates with the number of points contained in it.

Figure 20 shows the outputs of two algorithms for the Unit Circle dataset. The left shape belongs to the Mapper algorithm, and the right shape belongs to the FM algorithm. They are the same shape, structure, number of nodes, and number of edges. The colors and sizes of the nodes are relatively similar. The information about nodes, edges, and samples is summarized in Table 7.

The overlapping threshold is adjusted so that the output of both algorithms, Mapper and FM, is similar in the topological sense. Then, its value is recognized in the range from 0.10 to 0.19. After that, the silhouette coefficient scores are calculated in some representative cases, and the results are reported in Table 8. In this case, it can be seen that the scores of the FM algorithm are always higher, so their mean is also higher. This phenomenon is inevitable because the FCM algorithm updates the membership degrees and the cluster centroids to cover the filter range. The cluster centroids divided the cover interval reasonably based on the density of the data

Table 8: The silhouette coefficient scores for the Unit Circle dataset.

Mapper	OVERLAPPING PERCENTAGE	SILHOUETTE COEFFICIENT SCORE	
	50%	0.128	
FM	OVERLAPPING THRESHOLD	SILHOUETTE COEFFICIENT SCORE	
	0.10	0.207	mean = 0.226
	0.11	0.214	
	0.12	0.214	
	0.13	0.219	
	0.14	0.227	
	0.15	0.229	
	0.16	0.229	
	0.17	0.229	
	0.18	0.240	
	0.19	0.247	

points. Overall, the FM algorithm is considered better because of the high silhouette coefficient scores.

3.3.2 Reaven and Miller Diabetes Dataset

The Reaven and Miller diabetes dataset resulted from a study performed at Stanford University in the 1970s [181]. In this, 145 non-obese adult patients were examined. Six quantities were reported for each patient, namely age, relative weight, fasting plasma glucose level, test plasma glucose level (a measure of glucose intolerance), plasma insulin during the test (a measure of insulin response to oral glucose), and steady-state plasma glucose (a measure of insulin resistance). Therefore, the Reaven and Miller diabetes dataset has six dimensions.

In 1979, Reaven and Miller applied the projection pursuit method to this dataset. They obtained a direct visualization of the three-dimensional shape of the data set, as shown in Figure 21. This visualization is described as "a boomerang with floppy wings and a fat middle" [88]. The central core represents normal patients, while two "flares" emanating from the main body represent patients with chemical diabetes and overt diabetes. In 2007, the Mapper algorithm was described as an automatic tool for clearly detecting two "flares" in this diabetes dataset effectively only with a simple function in statistics [13]. This algorithm quickly creates highlights and emerges rapidly because of its simplicity and effectiveness.

In this case, both algorithms were used to analyze the Reaven and Miller diabetes dataset. Their corresponding parameters are described in Table 9. The Euclidean distance is considered a metric for this dataset. The filter is a KDE function in statistics. The clustering algorithm is the single-linkage clustering that uses default parameters from the scikit-learn library. The

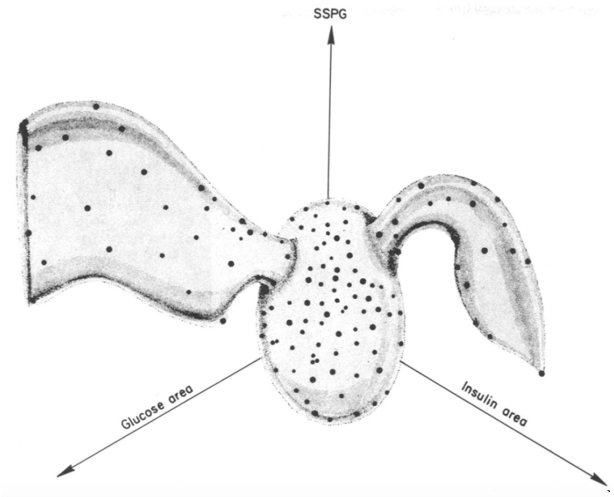
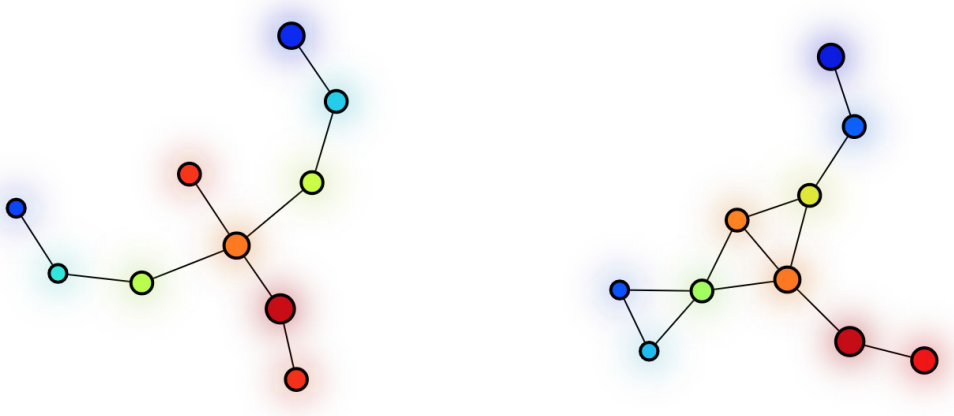


Figure 21: The three-dimensional visualization of the Reaven and Miller diabetes dataset.

Table 9: The parameter settings for the Reaven and Miller Diabetes dataset.

Mapper	FILTER FUNCTION	NUMBER OF INTERVALS	OVERLAPPING PERCENTAGE	CLUSTERING METHOD
	KDE	$N = 5$	$p = 50\%$	Single-Linkage
FM	FILTER FUNCTION	NUMBER OF INTERVALS	OVERLAPPING THRESHOLD	CLUSTERING METHOD
	KDE	$N = 5$	$\tau = 0.06$	Single-Linkage



(a) Mapper **(b)** FM

Figure 22: The outputs for the Reaven and Miller Diabetes dataset.

Table 10: The experimental results for the Reaven and Miller Diabetes dataset.

	NODES	EDGES	UNIQUE SAMPLES	TOTAL SAMPLES
Mapper	10	9	145	191
FM	10	12	145	215

node's color indicates the average lens values at points in the set represented by the node. The red expresses a high value, and the blue says a low value. The node's size indicates the number of points in the set represented by the node. This size positively correlates with the number of points contained in it.

Figure 22 shows the outputs of two algorithms for this diabetes dataset. The left shape provides the output of the original algorithm, and the right shape offers the output of the enhanced algorithm. They are the same in terms of shape and number of nodes. The colors and sizes of the nodes are relatively similar. Two "flares" that represent patients with chemical diabetes and overt diabetes appear in the output of each algorithm. The "flares" are not as symmetrical as those in the Mapper output for the FM algorithm. The connection between nodes is different in the two methods, especially concerning the left "flares". In the left shape, the connectivity happens for only two points to produce the edges. In contrast, this connectivity happens for three points to stimulate the closed lines with a triangular form in the right shape. It is caused by a difference in covering the filter range. The overlap between intervals in the FM algorithm cannot be consecutive and pairwise. The representation of the FM algorithm is very similar to the research results of Reaven and Miller because the density of points in two "flares" is varied, and some points in one "flare" are more scattered than in the other.

Table 10 summarizes the outputs' information about the nodes, edges, and samples. The FM algorithm creates more edges in this case as the overlap happens between many intervals.

The overlapping threshold is adjusted so that the output of both algorithms, Mapper and FM, is similar in the topological sense. Then, it receives a value from 0.056 to 0.061 to ensure that the above condition is guaranteed. After that, the silhouette coefficient scores are calculated, and Table 11 reports their results in some representative cases. These scores of the FM algorithm are all higher. The silhouette coefficient scores of both the algorithms are very near zero and have negative values, demonstrating overlap between clusters and wrong cluster assignment for some data points, respectively. The FM algorithm is assessed to be better in this experiment with a high score in the silhouette coefficients.

3.3.3 NKI Breast Cancer Dataset

In 2013, the NKI breast cancer dataset [182] was used in a TDA widely cited paper [17] to prove the usefulness of topology in extracting insights from the shape of complex data. Besides survival information, the NKI dataset consists of gene expression levels extracted from 272 tumors and is analyzed using about 1500 most varying genes. The Mapper algorithm demon-

Table 11: The silhouette coefficient scores for the Reaven and Miller Diabetes dataset.

Mapper	Overlapping Percentage	Silhouette Coefficient Score	
	50%	−0.036	
FM	Overlapping Threshold	Silhouette Coefficient Score	
	0.056	−0.029	mean = −0.028
	0.057	−0.029	
	0.058	−0.029	
	0.059	−0.026	
	0.060	−0.026	
	0.061	−0.026	

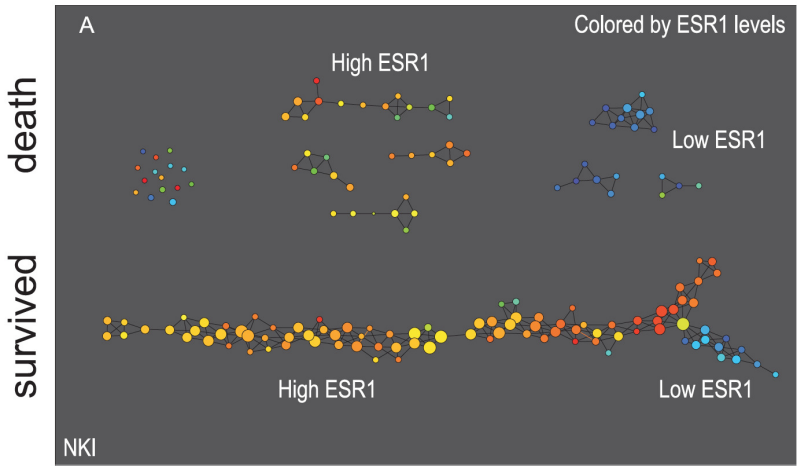


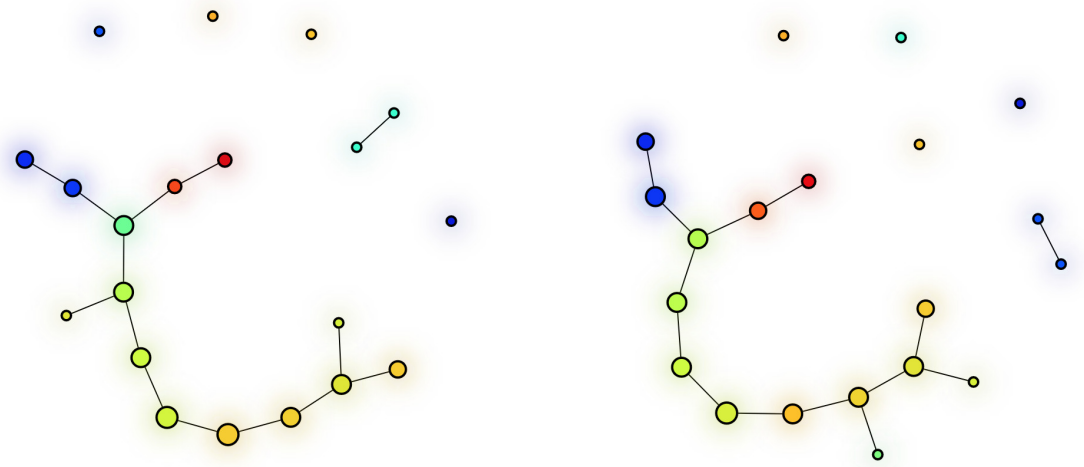
Figure 23: The networking visualization has a structure shaped like a horizontal letter Y along with several separate components.

strated its effectiveness in stratifying patients more finely than standard clustering methods. Suitably identifying cancer subtypes is complex in the medical field since sub-populations can be small and their relationships very difficult. Moreover, it is essential to identify interesting patient sub-groups for targeted therapy to save their lives.

Through the NKI breast cancer dataset, Lum et al. used the Mapper algorithm to identify unique patient subsets related to the estrogen receptor gene (ESR1) [17]. In the medical field, the expression level of ESR1 positively correlates with improved prognosis, given that patients with a high ESR1 level are likely to respond well to standard therapies. The Mapper algorithm used its geometric representation to visualize this dataset using gene expression and survival information. It proves that sub-groups still do not respond well to therapy among these high ESR1 patients. In addition, although low ESR1 levels strongly correlate with poor prognosis, there are still sub-groups with high survival over the years among these low ESR1 patients.

Table 12: The parameter settings for the NKI Breast Cancer dataset.

Mapper	Filter Function	Number of Intervals	Overlapping Percentage	Clustering Method
	L-Infinity Centrality	$N = 10$	$p = 30\%$	Single-Linkage
FM	Filter Function	Number of Intervals	Overlapping Threshold	Clustering Method
	L-Infinity Centrality	$N = 10$	$\tau = 0.08$	Single-Linkage



(a) Mapper **(b)** FM

Figure 24: The outputs for the NKI Breast Cancer dataset.

These results are proven through the network with a structure shaped like a horizontal letter Y and several separate components in Figure 23.

Both algorithms are used to analyze the NKI breast cancer dataset in this situation. Their corresponding parameters are described in Table 12. The correlation distance is considered a metric for this dataset. The filter is the L-infinity centrality function that computes for every data point the maximal distance to any other data point in the dataset. The clustering algorithm is single-linkage clustering that uses default parameters from the scikit-learn library. The node's color indicates the average lens values at points in the set represented by the node. The red expresses a high value, and the blue says a low value. The node's size indicates the number of points in the set represented by the node. This size positively correlates with the number of points contained in it.

Figure 24 shows the outputs of two algorithms on this breast cancer dataset. The right shape indicates the FM output, and the left shape means the Mapper output. These shapes are the same concerning form, number of nodes, number of edges, and number of connected components. The colors and sizes of the nodes are relatively similar. In this experiment, although only

Table 13: The experimental results for the NKI Breast Cancer dataset.

	NODES	EDGES	UNIQUE SAMPLES	TOTAL SAMPLES
Mapper	20	14	272	365
FM	20	14	272	398

Table 14: The silhouette coefficient scores for the NKI Breast Cancer dataset.

<table>
<tr><td rowspan="2">Mapper</td><td>OVERLAPPING PERCENTAGE</td><td colspan="2">SILHOUETTE COEFFICIENT SCORE</td></tr>
<tr><td>30%</td><td colspan="2">−0.119</td></tr>
<tr><td rowspan="14">FM</td><td>OVERLAPPING THRESHOLD</td><td colspan="2">SILHOUETTE COEFFICIENT SCORE</td></tr>
<tr><td>0.10</td><td>0.207</td><td rowspan="13">mean = −0.099</td></tr>
<tr><td>0.07</td><td>−0.095</td></tr>
<tr><td>0.08</td><td>−0.094</td></tr>
<tr><td>0.09</td><td>−0.099</td></tr>
<tr><td>0.10</td><td>−0.100</td></tr>
<tr><td>0.11</td><td>−0.095</td></tr>
<tr><td>0.12</td><td>−0.100</td></tr>
<tr><td>0.13</td><td>−0.100</td></tr>
<tr><td>0.14</td><td>−0.100</td></tr>
<tr><td>0.15</td><td>−0.100</td></tr>
<tr><td>0.16</td><td>−0.100</td></tr>
<tr><td>0.17</td><td>−0.100</td></tr>
</table>

one filter was used to process instead of two filter functions, as in [17], the outputs still have a structure shaped like a letter Y with several separate components. Both the Mapper and the FM algorithms detect special patient sub-groups. The first sub-group, which has high survival, although low ESR1, is shown on the branch colored blue of the letter Y structure. The second sub-group, which has high ESR1 but does not respond well to therapy, is shown in the orange disconnected components. The clear quantitative descriptions of nodes, edges, and samples are summarized in Table 13.

The overlapping threshold is adjusted so that the output of both algorithms, Mapper and FM, is similar in the topological sense. Then, its value is limited from 0.07 to 0.17, while the overlapping percentage of the Mapper algorithm is fixed. After that, the silhouette coefficient scores are calculated in some representative cases, and the results are reported in Table 14 with some specific values. All of them are also higher than the score in the original method. Once again, the scores of both algorithms are also near zero and negative. With the advantages inherited from fuzzy clustering in covering the filter range, the FM algorithm has good scores in the silhouette coefficient, so it clusters data better than the Mapper algorithm.

Overall, the following observations are recognized concerning the experimental results on the three real-world datasets.

(1) Firstly, from the topological standpoint, the output of the FM algorithm is somewhat similar to that of the Mapper algorithm in cases when the respective parameters are well-chosen.

(2) Secondly, the FM algorithm usually generates higher total samples than the Mapper algorithm. This phenomenon can be explained entirely at the phase of covering the filter range in each algorithm. In the Mapper algorithm, each interval only intersects with the consecutive intervals. But, because of the automation in dividing cover intervals in the FM algorithm, each interval can cross with more intervals and cannot necessarily be consecutive.

(3) Finally, the FM algorithm has a good clustering evaluation based on the silhouette coefficient score because of inheriting some valuable properties from the overlapping clustering.

3.4 Discussion

In this chapter, the Mapper algorithm was enhanced to the FM algorithm by applying the advantages of the FCM clustering algorithm. It is optimized for effectively choosing the suitable cover for the filter range. The cover choice develops towards the automatic division of intervals and natural overlap with random percentages. The excellent ability of the FM algorithm is demonstrated through experiments on real-world datasets and corresponding comparisons to the original method. The FM can generate outputs similar to those of the Mapper algorithm from the topological standpoint, although the cover choices on these algorithms are different. Furthermore, this novel algorithm produces well-clustered results based on the silhouette coefficient score. This good result is caused by sensibly dividing the filter range into intervals based on the density and distribution of data points.

This development represents the first step of different approaches to parameter choice optimization for the Mapper algorithm in the future. This algorithm is quite sensitive to the choices of filter and cover, so optimizing the parameters to achieve a better output is still an exciting and open problem. Moreover, in the FM algorithm, an overlapping threshold is only found with positive results. The best choice for this parameter is not still fully solved. Furthermore, future research could also study the general FM algorithm, which has an output that is multi-dimensional simplicial complexes. We expect that these problems will be addressed soon.

4 Shape FCM Algorithm

In Chapter 4, a novel algorithm, called *Shape FCM*, is constructed based on the FCM algorithm with the exclusive properties of the Mapper algorithm. This algorithm can not only exhibit the same clustering ability as the FCM algorithm, but also reveal some relationships through visualizing the global data shape supplied by the Mapper algorithm. It is presented with convincing arguments, well-proved evidence, and precise experiments. The algorithm performance is demonstrated through a comparative analysis involving the original algorithm, Mapper, and the other fuzzy set-based improved algorithm, FM. Four synthetic and real-world data datasets, including Unit Circle, Two Concentric Circles with Noise, 3D Trefoil Knot, and Reaven & Miller Diabetes. The comparison is conducted concerning output shape in the topological sense, clustering stability, and internal evaluation. Moreover, the SFCM algorithm also shows its potential to big data by testing on high-dimensional datasets with acceptable results.

Overall, the contributions of Chapter 4 can be summarized as follows:

[C4.1] Proposing the SFCM algorithm with two simultaneous possibilities for mining data: fuzzy clustering with the FCM algorithm and shape detecting as with the Mapper algorithm.

[C4.2] Demonstrating that the SFCM algorithm can generate similar outputs from the topological standpoint as the Mapper and FM algorithms.

[C4.3] Reporting well-clustered results based on the SFCM algorithm's clustering stability and internal index in most experimental cases compared to the Mapper and FM algorithms.

[C4.4] Demonstrating that the SFCM algorithm can visualize highly complex data in a simple, meaningful, and informative form with potential applicability to big data.

Moreover, the rest of this chapter is organized as follows. Section 4.1 presents the motivation for proposing the SFCM algorithm. Section 4.2 describes the algorithm carefully with visual illustrations. The experimental results on four real-world datasets are reported clearly in Section 4.3 based on the comparison between this proposed algorithm and two previous algorithms. Finally, the conclusions in Section 4.4 review what has been achieved to orientate and develop future researches on the SFCM algorithm.

4.1 Motivation

Clustering is an effective data mining technique. It divides data into groups of similar objects, known as clusters, that are meaningful and useful [34]. Objects in the same group are more similar in a certain sense than others in other groups. There are two types of clustering, exclusive clustering, and overlapping clustering. In exclusive clustering, each object belongs

to exactly one cluster. Each object can belong to two or more clusters in overlapping clustering depending on the membership function. Membership degree, assigned to each object, indicates the degree to which this object belongs to each cluster. The fuzzy set concept introduced by Zadeh in 1965 [38] is considered the inspiration for overlapping clustering. Fuzzy set theory has become an increasingly useful tool to describe situations in which the data are imprecise or vague, such as linguistics [183, 39, 184], decision-making [41, 185, 186], web mining [44, 187, 188], frequent itemset mining [189, 190, 191], bioinformatics [192, 193, 194], machine learning [195, 196, 197], and so on.

It is worth noting that the Mapper algorithm is still a fuzzy clustering algorithm, from a theoretical perspective, with a visualization ability to extract the shape summary of data. The algorithm results are still very sensitive to the choice of the resolution parameters on the cover. Covering the filter range by equal intervals with the same overlapping percentage is also a weakness in the cover choice. Moreover, determining which filter to be chosen is another problem that needs to be considered when applying the Mapper algorithm in practice. Inspired by the essence of this algorithm in the trend of optimizing the choice of parameters, a novel algorithm, called the SFCM, is proposed based on the two algorithms, FCM and Mapper. The SFCM algorithm is constructed on the foundation of a bright and significant representative in overlapping clustering by carefully combining with the unique advantages of TDA. This algorithm brings two simultaneous data possibilities: fuzzy clustering with the FCM algorithm and shape detection with the Mapper algorithm. It is proposed as a summarization technique to transform large and complex data into informative representations and provide interactive visualizations for their exploration. In particular, from the TDA perspective, the SFCM algorithm significantly reduces the dependence on the parameters. Like the original methods, this approach is expected to solve practically and robustly some problems encountered in many fields, especially in bioinformatics and neuroscience [59, 20, 21, 198].

The SFCM algorithm is introduced as a fuzzy clustering algorithm for extracting meaningful and straightforward descriptions of high-dimensional datasets. This algorithm is constructed by carefully combining the advantages of the two algorithms, FCM and Mapper.

(1) Firstly, concerning the FCM algorithm, the SFCM algorithm has not only entirely inherited its advantages in clustering, but also becomes powerful in terms of the abilities of qualitative analysis, simplification, and visualization of high-dimensional datasets.

(2) Secondly, compared to the Mapper algorithm, the parameter choice of the SFCM algorithm is quite simple, as it only depends on two items instead of four being encountered in the original method

Therefore, the development and improvement of both component algorithms can be used in this new algorithm to increase its operating performance in specific ways.

In conclusion, combining clustering and extracting shape information in data mining is a relatively new, promising, and feasible idea. This combination is conducted on the famous

Table 15: The comparison between algorithms in the corresponding steps.

Step	Mapper algorithm	SFCM algorithm
1	Projecting all data points in $\mathbb{X}$ to $\mathbb{R}$ by using the filter f.	
2	Covering the filter range $f(\mathbb{X})$ by a cover $\mathcal{I}$ that depends on the following resolution parameters: – N regular intervals, – Same overlapping percentage p.	Covering the filter range by a cover $\mathcal{I}$ that is obtained by the FCM algorithm and depends on the following resolution parameters: – N irregular intervals, – Random overlapping percentage based on the threshold τ.
3	Decomposing the pre-image $f^{-1}(I_i)$ of each cover interval I_i into clusters $C_{i,1}$, $C_{i,2}$, $C_{i,3}$, ..., C_{i,n_i} by using the clustering algorithm $\mathcal{C}$ or partitioning connected components based on the distance metric d.	
4	Constructing the simplicial complex $\mathcal{G}$ defined by clusters C_ij and their intersections.	

representative of each field, overlapping clustering and TDA. How are two well-known algorithms, FCM and Mapper, combined? What are the similar and different points of the perfect combined method with the original component methods? The following section discussed these questions.

4.2 Description

In the same way, as with the Mapper algorithm, the SFCM algorithm uses a discrete dataset $\mathbb{X}$ as its *input*. Due to the combination of two algorithms, this algorithm starts working with four user-defined *parameters* as follows:

– Number of clusters C.

– Fuzzification exponent m whose value has been often chosen the fixed value $m = 2.0$ based on the previous empirical studies.

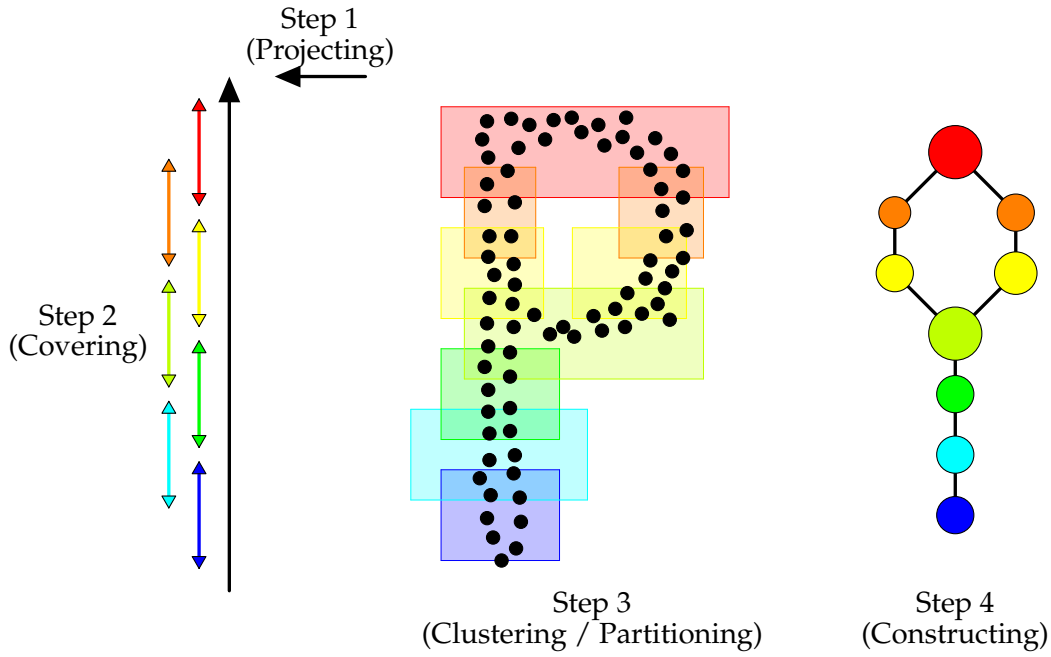
– The termination criteria, $\varepsilon \in (0; 1)$ or k_{max}.

– The overlapping threshold τ that decides the overlap between clusters by comparing the membership degree of each data point.

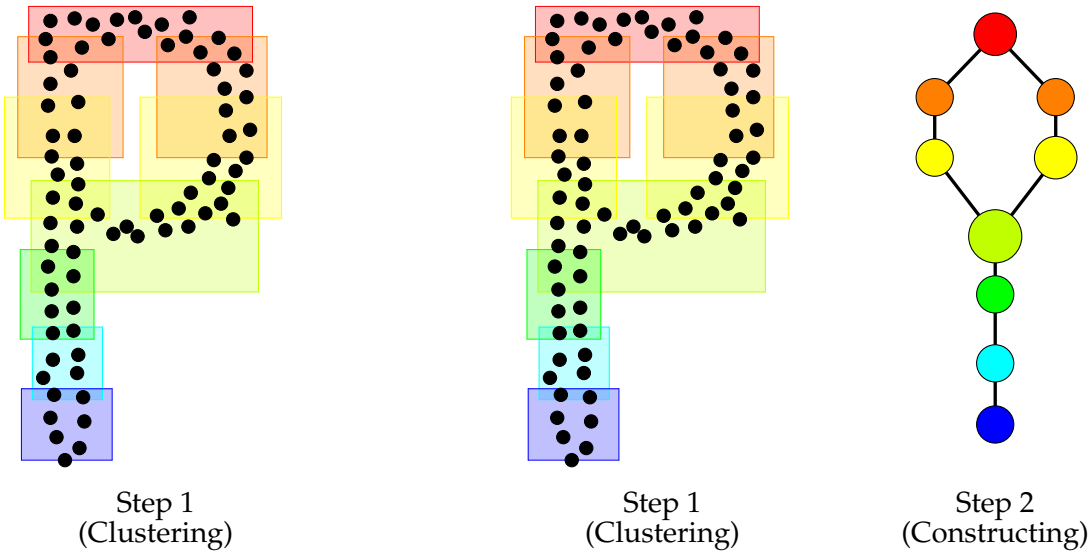
The SFCM algorithm also creates a geometric representation of the dataset $\mathbb{X}$ that consists of nodes and edges, as in the case of the Mapper algorithm. The number C of clusters and the overlapping threshold τ called the *resolution parameters*, affect the output result. A node

Table 16: The description of the SFCM algorithm.

Input	Dataset $\mathbb{X}$ with finite elements.
Parameters	– Number of clusters C, – Fuzzification exponent m that is often set to 2.0, – Termination criteria: $\varepsilon \in (0;1)$ or $k_{\max}$. – The overlapping threshold τ.
Method	1: Clustering on $\mathbb{X}$ by using the FCM algorithm: 1.1: $k = 0$. 1.2: Initializing the fuzzy partition matrix $U^{(0)}$. 1.3: **repeat** 1.3.1: Calculating the cluster centroid matrix $\mathcal{C}^{(k)} = (v_j)_{1\times C}$ by using the following formula: $$\forall j = 1, 2, \ldots, C, v_j^{(k)} = \frac{\sum_{i=1}^{n} (u_{ij})^m x_i}{\sum_{i=1}^{n} (u_{ij})^m}.$$ 1.3.2: Updating the fuzzy partition matrix $U^{(k)} = (u_{ij})_{n\times C}$ by using the following formula: $$\forall i = 1, 2, \ldots, n, \forall j = 1, 2, \ldots, C, u_{ij}^{(k)} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|}\right)^{\frac{2}{m-1}}}.$$ 1.3.3: $k = k + 1$. 1.4: **until** $\max_{i,j} \left\{ \left| u_{ij}^{(k+1)} - u_{ij}^{(k)} \right| \right\} < \varepsilon$ or $k = k_{\max}$. 1.5: Using the overlapping threshold τ to identify C clusters from the fuzzy partition matrix U. 4: Constructing the simplicial complex $\mathcal{G}$ defined by clusters C_i and their intersections.
Output	Simplicial complex $\mathcal{G}$ as a geometric representation of $\mathbb{X}$.



(a) Mapper algorithm



(b) FCM algorithm

(c) SFCM algorithm

Figure 25: The illustration of algorithms on a sampled point cloud with a noisy P structure.

represents each cluster, and an edge shows each connection between two clusters. The number of nodes positively correlates with the number of clusters in the output shape, and the connectivity negatively correlates with the overlapping threshold. The color and the size are the two main characteristics of nodes. The color usually indicates the average of the colored values of points, while the size usually instructs the number of points. The blue and red sequentially display the minimal and maximal values, respectively. The colors ranging from blue to red express the colored values ranging from low to high.

The SFCM algorithm is deployed through two main processes:

(1) **Step 1.** In the first process, the FCM algorithm organizes the points of the dataset $\mathbb{X} = \{x_1, x_2, \ldots, x_n\}$ into C clusters correspoding to centroids $\mathcal{C} = \{v_1, v_2, \ldots, v_C\}$ with the fuzzy partition matrix U determined by the membership degrees of the points for each cluster. The main task of this process is to iteratively minimize the following *objective function*:

$$\sum_{i=1}^{n}\sum_{j=1}^{C}(u_{ij})^m \|x_i - v_j\|^2, \tag{22}$$

where the membership degree u_{ij} of the point x_i in the cluster j is constrained as follows:

$$\forall i = 1, 2, \ldots, n, \forall j = 1, 2, \ldots, C, u_{ij} \in [0; 1], \tag{23}$$

and

$$\forall i = 1, 2, \ldots, n, \sum_{j=1}^{C} u_{ij} = 1. \tag{24}$$

The fuzzification exponent m is any real number greater than 1 and is often set to 2.0. The norm $\|.\|$ is a Euclidean metric expressing dissimilarity between arbitrary points and a given centroid.

The fuzzy partition is carried out through iterative minimization of the objective function through updating the cluster centroids v_j and the membership degrees u_{ij} by the following formulas:

$$\forall j = 1, 2, \ldots, C, v_j = \frac{\sum_{i=1}^{n}(u_{ij})^m x_i}{\sum_{i=1}^{n}(u_{ij})^m}, \tag{25}$$

and

$$\forall i = 1, 2, \ldots, n, \forall j = 1, 2, \ldots, C, u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|}\right)^{\frac{2}{m-1}}}. \tag{26}$$

The FCM algorithm stops when one of the termination criteria is satisfied:

$$\max_{i,j}\left\{\left|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right|\right\} < \varepsilon \in (0; 1), \tag{27}$$

or

$$k = k_{\max}, \tag{28}$$

The clusters then serve as the covers for $\mathbb{X}$, and the overlapping threshold τ is used to express the degree of overlap between any two members of the covers. Using a threshold for specific purposes is a widespread technique in fuzzy data mining [178, 179]. A data point belongs to a cluster if its membership degree for this cluster is greater than or equal to the threshold τ. Therefore, it can belong to one or more clusters.

(2) **Step 2.** In the second process, the nerve of the refined cover defined by the clusters on data $\mathbb{X}$ is constructed as the Mapper algorithm does. It is necessary to note that the refined cover is generated from the overlapping clusters in the cover on filter range by separating them into single connected components. The node v_i represents for each component C_i. Every pair of nodes corresponding to two clusters C_1 and C_2 is connected by an edge if and only if their intersection $C_1 \cap C_2$ is not empty. The color and the size are the two main characteristics of nodes. The color usually indicates the average of the colored values of points, while the size usually instructs the number of points. The blue and red sequentially display the minimal and maximal values, respectively. The colors ranging from blue to red express the colored values ranging from low to high.

The SFCM algorithm is described in detail with the steps in Table 16. It is compared with the original algorithm, specifically in Table 15. As an example, the SFCM algorithm is annotated on a point cloud sampled with the noisy P structure in Figure 25 with two-component algorithms, FCM and Mapper. In this case, the point cloud is first covered by the nine clusters obtained by the FCM algorithm. The graph is then composed of vertices representing the clusters and edges representing the connections between the two non-intersecting clusters. The values of the data points color the nodes through the projection on the vertical line. The high value is red, and the low value is blue.

The membership degree u_{ij} of point x_i to the cluster C_j belongs to the closed interval $[0; 1]$ and $\sum_{j=1}^{C} u_{ij} = 1$ for all $1 \leq i \leq n$. Let

$$T_0 = \min_{i,j} u_{ij}, \tag{29}$$

and

$$T_1 = \min_{i} \max_{j} u_{ij} \tag{30}$$

If $\tau \leq T_0$, for all $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, C\}$, all points x_i belong to all clusters C_j, so the output is a complete graph. The outputs have the same presentations for all $0 \leq \tau \leq T_0$. If $\tau > T_1$, there exist points that are not in any cluster. Therefore, the potential values of τ are limited to a close interval $[T_0; T_1]$. The point x_i belongs to the cluster C_j if and only if

$u_{ij} = \max_k u_{ik}$ in case $\tau \geq T_1$. This condition ensures that each data point belongs to at least a single cluster.

Theorem 1 *If $G_\tau = (V, \Sigma_\tau)$ is the output graph of the SFCM algorithm corresponding to threshold τ, then for all $\tau, \tau' \in [T_0; T_1]$,*

$$\tau' \leq \tau \Longrightarrow G_\tau \subseteq G_{\tau'} \tag{31}$$

Proof Because G_τ and $G_{\tau'}$ have the same vertices set V, $G_\tau \subseteq G_{\tau'} \Longleftrightarrow \Sigma_\tau \subseteq \Sigma_{\tau'}$. So, it is enough to just prove $\Sigma_\tau \subseteq \Sigma_{\tau'}$ in the condition $\tau \geq \tau'$.

For the threshold τ, for all $e \in \Sigma_\tau$, exist $v_i, v_j \in V$, such that e connects the two vertices v_i and v_j. Then, there exists the least $x_k \in C_i^\tau \cap C_j^\tau$ in which C_i^τ is the ith cluster corresponding to the threshold τ.

$$\begin{aligned} x_k \in C_i^\tau \cap C_j^\tau \Longrightarrow \begin{cases} x_k \in C_i^\tau \\ x_k \in C_j^\tau \end{cases} &\Longrightarrow \begin{cases} u_{ki} \geq \tau \\ u_{kj} \geq \tau \end{cases} \\ &\Longrightarrow \begin{cases} u_{ki} \geq \tau' \\ u_{kj} \geq \tau' \end{cases} \Longrightarrow \begin{cases} x_k \in C_i^{\tau'} \\ x_k \in C_j^{\tau'} \end{cases} \Longrightarrow x_k \in C_i^{\tau'} \cap C_j^{\tau'}. \end{aligned} \tag{32}$$

This inference implies that for the threshold τ', e connects the two vertices v_i and v_j. So, we have $e \in \Sigma_{\tau'}$. Therefore, $\Sigma_\tau \subseteq \Sigma_{\tau'}$. ∎

Theorem 2 *The non-decreasing sequence of the threshold values generates the filtered sequence of homology groups.*

Proof Let $0 \leq \cdots \leq \tau_{k-1} \leq \tau_k \leq \tau_{k+1} \leq \cdots \leq 1$ be a non-decreasing sequence of the threshold values. According to Theorem 1, there is a non-decreasing sequence of the output graph generated by the SFCM algorithm as follows:

$$\cdots \subseteq G_{\tau_{k+1}} \subseteq G_{\tau_k} \subseteq G_{\tau_{k-1}} \subseteq \cdots \tag{33}$$

This sequence is a filtered simplicial complex. Therefore, according to the algebraic topology, there is a sequence of homology groups that generates the filtered simplicial complex as follows:

$$\ldots \subseteq H\left(G_{\tau_{k+1}}\right) \subseteq H\left(G_{\tau_k}\right) \subseteq H\left(G_{\tau_{k-1}}\right) \subseteq \cdots \tag{34}$$

∎

Note that there are only finitely many thresholds where the structure of the output graph G_τ can change. As such, there is a finite non-decreasing sequence of the thresholds as follows:

$$0 \leq T_0 = \tau_0 \leq \tau_1 \leq \tau_2 \leq \cdots \leq \tau_l = T_1 \leq 1 \tag{35}$$

According to Theorem 1, this yields the sequence of homology groups as follows:

$$H\left(G_{\tau_l}\right) \subseteq H\left(G_{\tau_{l-1}}\right) \subseteq \cdots \subseteq H\left(G_{\tau_1}\right) \subseteq H\left(G_{\tau_0}\right) \tag{36}$$

Since the algorithm's output is a graph, which is a simplicial complex with its dimension not greater than one, all homology groups in Equation (34) thus have a dimensionality not greater than one as well. The SFCM algorithm can be extended entirely to the general form when the nerve of the cover by the clusters is constructed in higher dimensions. These theoretical results are the same as using the Čech complex or the Vietoris-Rips complex to build the simplicial complex on the dataset. These create the premise for analyzing data using the practical tools of persistent homology.

Additionally, the time complexity of the SFCM algorithm can be assessed through two-component algorithms, FCM and Mapper.

Theorem 3 *The time complexity of the SFCM algorithm is $\mathcal{O}\left(kcNn^2\right)$, where k, C, n, and N are the numbers of iterations, clusters, objects, and attributes, respectively.*

Proof In the first process, for each loop, the calculations in Equations (26), (25), (22) require $\mathcal{O}\left(cnN\right)$, $\mathcal{O}\left(cnN^2\right)$, $\mathcal{O}\left(c^2n\right)$ operations, respectively [199]. So, the time complexity of the FCM algorithm is $\mathcal{O}\left(k\left(cnN + cnN^2 + c^2n\right)\right) \rightarrow \mathcal{O}\left(kcnN^2\right)$. In the second process, it takes at most $\mathcal{O}\left(k\left(k-1\right)\right)$ steps to create the connections between nodes. Overall, the time complexity of the SFCM algorithm is $\mathcal{O}\left(kcnN^2 + k\left(k-1\right)\right) \rightarrow \mathcal{O}\left(kcnN^2\right)$. ∎

In brief, the SFCM algorithm is not only a fuzzy clustering algorithm but also is equipped with the ability to extract the shape information of a high-dimensional dataset. The FCM algorithm takes advantage of being efficient to create the cover for the whole space. This property offers a new and sound idea of high feasibility because it produces clusters naturally based on the density and distribution of data points. The number of clusters helps the users proactively decide the simplicity or complexity in the output visualization. Overall, with the mentioned highlights, the SFCM algorithm can visualize highly complex data in a meaningful and straightforward form by a proper parameter selection depending on users' perspectives. Beyond clustering capabilities, this algorithm can also be understood as a method for feature compression and used for big data visualization with lower-dimensional approximate representation in the most understandable and informative form. It can be said that the clustering connects with the visualization naturally, efficiently, and ideally by providing an excellent theoretical foundation for simplifying large and complex data while preserving their core structures. Table 17 compares of the four related algorithms, namely the Mapper, FM, FCM, and SFCM.

Through some illustrations on artificial data sets, both algorithms, SFCM and Mapper, generate similar results. It should be noted that this is only a subjective conjecture based on the shapes and comments from these datasets. Some experiments are needed to proceed on real

Table 17: The comparison of properties between related algorithms.

	Number of Parameters	Type of Clustering	Clustering Ability	Shape Extracting Ability
Mapper	3	Overlapping	Yes	Yes
FM	3	Overlapping	Yes	Yes
FCM	3	Overlapping	Yes	No
SFCM	2	Overlapping	Yes	Yes

datasets to confirm this assertion. Moreover, the FM algorithm is also a fuzzy set-based method similar to the two mentioned algorithms. Therefore, the following section discusses the real datasets' characteristics and shows experimental evaluations to compare the results among the three algorithms, SFCM, FM, and Mapper.

4.3 Experiment

Many software packages use the Mapper algorithm as a theoretical framework and are freely available in MATLAB, Python, R, and Spark [35]. In this implementation, the KeplerMapper, a free library implementing the Mapper algorithm in Python by Saul and Veen et al. [112, 113], has been used to conduct the experiments. The authors of this library have provided a fast and flexible tool with a user-friendly interface to the scientific community. This package recently developed and improved as a library in the Scikit-TDA project [126] that provides TDA Python tools in widely usable and easily approachable forms. The code in the open-source codebase of the KeplerMapper is used to extend and modify to create source code for the novel algorithms, FM and SFCM. Besides, the SciKit-Fuzzy package is also used to implement the experiments for the FCM algorithm. It is a library of fuzzy logic algorithms in the SciPy Stack by the Python computing language. If nothing changes, the termination criteria, reviewed in this section, are set as a constant $k_{max} = 10,000$ and $\varepsilon = 0.0001 \in (0; 1)$.

In this section, the efficiency of the SFCM algorithm is evaluated on three aspects: the output shape from the topological standpoint, the clustering stability with the matching coefficient, and the internal index with the silhouette coefficient. To do this, the experiments are implemented transparently on the four real-world datasets with detailed descriptions. The Euclidean distance is considered a metric for these datasets. Moreover, to prove the working ability on big data of this algorithm, some experimental runs are also conducted on large datasets with high dimensions.

All three algorithms, Mapper, FM, and SFCM, are processed sequentially on each real-world dataset for the output shape.

(1) The parameters of the Mapper algorithm, such as the filter f, the number N of intervals, overlapping percentage p, and clustering algorithm $\mathcal{C}$, are set as same as in the previous well-known works.

(2) The parameters of the FM algorithm concerning the filter f, number N of intervals, and clustering algorithm $\mathcal{C}$, are chosen the same as the Mapper algorithm. A tested value assigns the overlapping threshold τ for achieving a similar shape to the Mapper algorithm from the topological standpoint.

(3) Finally, the parameters of the SFCM algorithm are chosen to show that it is possible to generate shapes that are pretty similar to those of the Mapper algorithm from the topological standpoint. The number N of clusters is set the same as those in the graph of the Mapper algorithm. The overlapping threshold τ is assigned with a specific value to get a similar shape to the Mapper algorithm from the topological standpoint.

In these experiments, the overlapping percentage p and the overlapping threshold τ are chosen to have exactly two decimal places.

Clustering stability is considered the main feature to confirm the validity of the sample-based algorithms [200]. In practice, it is widely used for optimizing the algorithm's parameters. However, the theoretical analysis of this notion is quite limited, with only the Ben-Hur [201], Luxburg [202], and Shai Ben-David [200] research works. These experiments discuss clustering stability following the Ben-Hur [201] view through the matching coefficient.

Given a dataset $\mathbb{X} = \{x_1, x_2, \ldots, x_n\}$ with n points. The clustering with k clusters on a dataset $\mathbb{X}$ is a labeling function $\mathcal{L} : \mathbb{X} \longrightarrow \{1, 2, \ldots, k\}$ that assigns labels to all points of $\mathbb{X}$. A clustering algorithm is a procedure that takes a set $\mathbb{X}$ of points as input and outputs a clustering of $\mathbb{X}$. Each labeling function $\mathcal{L}$ is represented by a matrix M defined as follows:

$$M_{ij} = \begin{cases} 0 & \text{, if } x_i \text{ and } x_j \text{ in the same cluster} \\ 1 & \text{, otherwise} \end{cases} \tag{37}$$

The dot product between two labeling functions L_1 and L_2 is defined as the dot product between two respective representative matrices M^1 and M^2 as follows:

$$\langle L_1, L_2 \rangle = \left\langle M^1, M^2 \right\rangle \tag{38}$$

The dot product computes the number of pairs of points clustered together.

For two matrices M^1 and M^2 with $0 - 1$ entries, let n_{ij} be the number of entries such that the matrix M^1 has the value i and the matrix M^2 has the value j. The matching coefficient, which is a measure used for comparing the similarity of two matrices, in this case, is defined as the fraction of the number of matching entries over the total number of entries as follows:

$$MC\left(L_1, L_2\right) = \frac{n_{00} + n_{11}}{n_{00} + n_{01} + n_{10} + n_{11}} \tag{39}$$

This similarity measure can be expressed in terms of the dot product as follows:

$$MC\left(L_1, L_2\right) = 1 - \frac{\left\langle M^1 - M^2, M^1 - M^2 \right\rangle}{n^2} \tag{40}$$

The clustering stability of all algorithms is evaluated based on the matching coefficient for each dataset. It is important to note that there is no standard procedure to determine clustering stability, and a discussion of the subject can be found in [202]. We conduct these experiments to compute the matching coefficient through the sub-sampling of the data detailed in [201]. The clustering stability is computationally estimated by the method of the k-fold cross-validation [203] as follows:

(1) Splitting the dataset into k sub-samples by choosing $m, k \in \mathbb{N}$ such that $n = km$ and removing the m points from $m\left(i-1\right)+1$ to mi for each $1 \leq i \leq k$, then obtaining k sub-samples with $\left(k-1\right)m$ points in each sample.

(2) Computing the matching coefficient between the clustering of each pair of sub-samples, on the $\left(k-2\right)m$ points of their intersection.

(3) Averaging the matching coefficients restricted to the sub-samples by summing the coefficients and dividing it by $C_k^2 = k\left(k-1\right)$.

The silhouette coefficient is a validation index of consistency within clusters of data [180]. A silhouette of a data point measures the clustering quality of the point. It compares the average distance of the point to all elements in the same cluster with its average distance to all elements in other clusters. The average computed for all points of a whole dataset measures the clustering quality.

This section uses a fuzzy version of the silhouette coefficient in the exclusive clustering [204] to evaluate these experiments. The silhouette $s\left(x_k\right)$ of a data point x_k is defined as:

$$s\left(x_k\right) = \frac{b_{pk} - a_{pk}}{\max\left\{a_{pk}, b_{pk}\right\}} \tag{41}$$

where a_{pk} is the distance from the point x_k to its nearest prototype v_p and b_{pk} is the distance from x_k to its second closest cluster prototype.

The silhouette $s\left(x_k\right)$ is in a closed interval $[-1, 1]$. The silhouette $s\left(\mathbb{X}\right)$ of a dataset $\mathbb{X}$ is the average for all points of the dataset as follows:

$$s\left(\mathbb{X}\right) = \frac{\sum\limits_{k=1}^{n}\left(u_{pk} - u_{qk}\right)s_k}{\sum\limits_{k=1}^{n}\left(u_{pk} - u_{qk}\right)} \tag{42}$$

where u_{pk} and u_{qk} are the first and the second largest elements of the k-th column in the fuzzy partition matrix, respectively. Good partitions are expected to bring greater values to $s\left(x_k\right)$

and thus to $s(\mathbb{X})$ than bad ones. So, the silhouette coefficient under the fuzzy version is also a sound maximization index [199].

All three algorithms, Mapper, FM, and SFCM, are also processed sequentially on each real-world dataset for the clustering stability and internal index.

- In the Mapper algorithm, the parameters are set as in the previous well-known experiments for each dataset. After that, the overlapping percentage p is changed in the condition that the output graphs have the same shape. The other parameters such as the filter, number of intervals, and clustering algorithm are fixed.

- In the FM algorithm, the parameters such as the filter, number of intervals, clustering algorithm are identified invariably as those in the Mapper algorithm for each dataset. The overlapping threshold τ is changed to guarantee the output graphs have the same shape as the Mapper algorithm from the topological standpoint.

- In the SFCM algorithm, the number of clusters is chosen to be the same as the number of nodes of the Mapper algorithm output. The overlapping threshold is selected to get a similar result to the Mapper algorithm from the topological standpoint of each dataset. The overlapping threshold τ is changed such that the output graphs have the same shape under the condition the number of clusters is fixed.

The matching coefficient is calculated using the k-fold cross-validation method for each case in which the overlapping percentage satisfies the above condition. At the same time, the silhouette coefficient is also calculated for each case corresponding to the matching coefficients. The matching score is the average of the matching coefficients for all instances of the overlapping resolutions. The outcome of each procedure is considered an approximation of the clustering stability of each algorithm. The silhouette score is also the mean of the silhouette coefficients of all cases of the overlap resolutions. The outcome of each procedure is considered an approximation of the internal index of each algorithm.

4.3.1 Unit Circle Dataset

The Unit Circle dataset consists of approximately 150 noisy points located on a unit circle. A unit circle is a circle of radius one and is centered at the origin in the Cartesian coordinate system. This dataset is one of the classic examples to illustrate the functioning of the Mapper algorithm [13, 93, 94, 35]. Figure 26 shows the visualization of this two-dimensional dataset.

All three algorithms, Mapper, FM, and SFCM, are used to analyze the Unit Circle dataset. The clear choice of the parameters for each algorithm is described in Table 18. The parameters for the Mapper algorithm are kept similar to those in the previous famous works. The DBSCAN algorithm is used the default settings from the scikit-learn package in this experiment. The parameters for the FM and SFCM algorithms are chosen to show that they can achieve the same results as the Mapper algorithm in terms of topology. The nodes in the graphs obtained by all

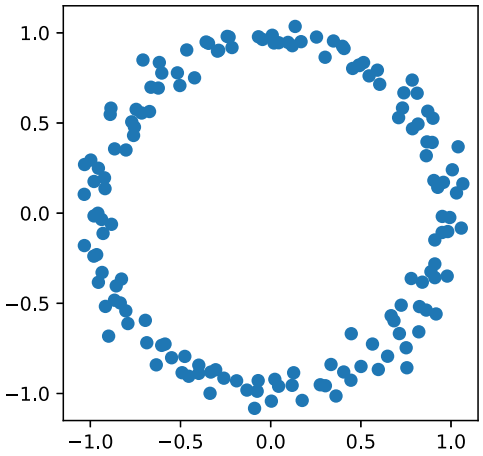


Figure 26: The visualization of the Unit Circle dataset.

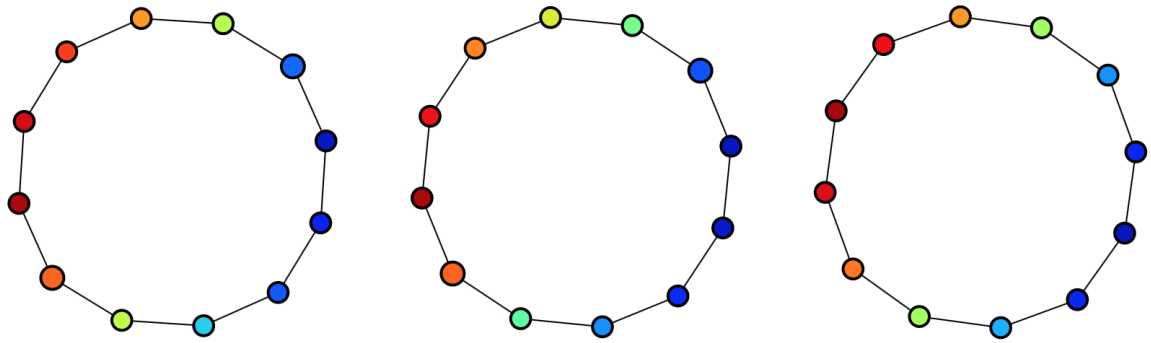
Table 18: The parameter settings for the Unit Circle dataset.

Mapper	Filter Function	Number of Intervals	Overlapping Percentage	Clustering Method
	Sum	$N = 7$	$p = 50\%$	DBSCAN
FM	Filter Function	Number of Intervals	Overlapping Threshold	Clustering Method
	Sum	$N = 7$	$\tau = 0.25$	DBSCAN
SFCM	Number of Clusters		Overlapping Threshold	
	$N = 12$		$\tau = 0.20$	

algorithms are colored by the same function, the mean of the coordinates of data points. The node's color indicates the average filter values at points in the set represented by the node. The red expresses a maximal value, and the blue says a minimal value. The node's size indicates the number of points in the set represented by the node. This size positively correlates with the number of points contained in it.

The experimental results of these algorithms on this dataset are shown in Figure 27. The left image belongs to the Mapper algorithm, the center image belongs to the FM algorithm, and the right image belongs to the SFCM algorithm. These images are pretty similar in terms of shape, structure, color, and size of nodes. This figure proves that the SFCM algorithm can create an output similar to the Mapper and FM algorithms from the topological standpoint.

In the paper [35], the FM algorithm can produce similar topological results as the Mapper algorithm can. Therefore, in this experiment, the SFCM algorithm's effectiveness is not only



(a) Mapper (b) FM (c) SFCM

Figure 27: The outputs for the Unit Circle dataset.

Table 19: The evaluation results for the Unit Circle dataset.

	MATCHING SCORE	SILHOUETTE SCORE
Mapper	0.861	0.122
FM	0.878	0.147
SFCM	0.883	0.213

evaluated with the Mapper algorithm, but also with the FM algorithm through the matching and silhouette scores.

Three algorithms, Mapper, FM, and SFCM, were alternately run to change the overlapping parameters such that there was no change in the output shapes in the condition that the other parameters were fixed. Some results are recognized for percentage and thresholds as follows:

- For the Mapper algorithm, the overlapping percentage p is varied from 42% to 50%.
- For the FM algorithm, the overlapping threshold τ is varied from 0.10 to 0.25.
- For the SFCM algorithm, the overlapping threshold τ is varied from 0.08 to 0.20.

For each algorithm, its effectiveness is evaluated through the matching and silhouette scores specifically as follows:

- The matching coefficient is calculated using the k-fold cross-validation method for each case in which the overlapping parameters satisfy the non-changed shape condition. In this experiment, k is set at 10 for sub-sampling the data, the same as the sampling ratio of 0.8 used in [201]. The matching score is the mean of the matching coefficients of all cases.
- The silhouette coefficient is also calculated for each case where the overlapping parameters do not change the output shape of the algorithms. The silhouette score is the mean of the silhouette coefficients of all cases.

The matching and silhouette scores of the three algorithms are reported in Table 19.

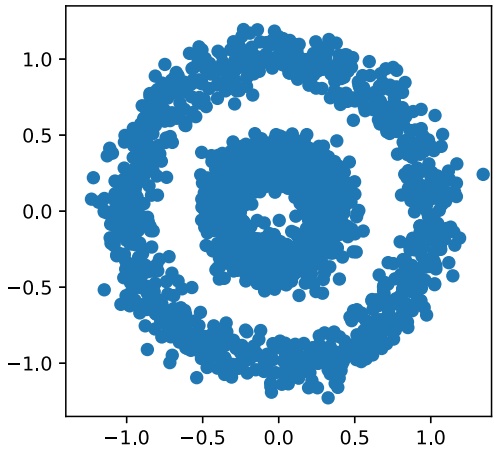


Figure 28: The visualization of the Two Concentric Circles dataset with Noise.

4.3.2 Two Concentric Circles Dataset with Noise

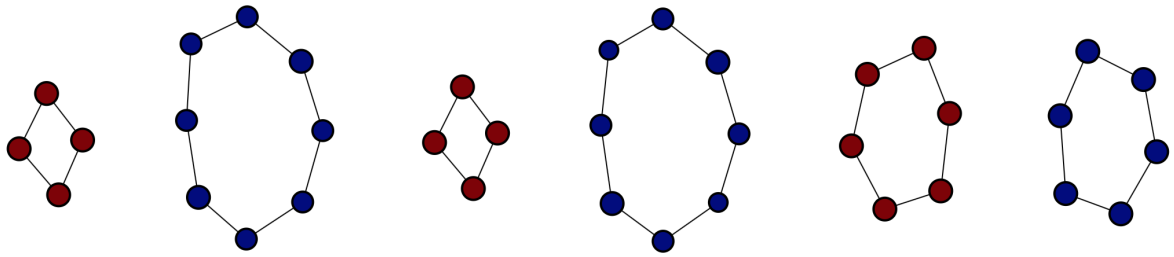
The Two Concentric Circles dataset with noise has approximately $2,000$ noise points that create a large circle containing a smaller circle. This dataset is often used to visualize clustering and classification algorithms [131]. The visualization of this dataset is presented in Figure 28 on the Euclidean plane.

This dataset is mined with the three algorithms, Mapper, FM, and SFCM. Their respective parameters are described in Table 20. The parameters for the Mapper algorithm are selected so that the output can detect the two circles' structures. The DBSCAN algorithm is also used with default settings from the scikit-learn package. The parameters for the FM and SFCM algorithms are chosen to show that both of them can achieve the same results as the Mapper algorithm in terms of topology. The nodes in the graphs obtained by these algorithms are colored by the same function, the mean of the coordinates of data points. The node's color indicates the average filter values at points in the set represented by the node. The red expresses a maximal value, and the blue says a minimal value. The node's size indicates the number of points in the set represented by the node. This size positively correlates with the number of points contained in it.

Figure 29 shows the results of the three algorithms obtained for this dataset. The left image belongs to the Mapper algorithm, the center image belongs to the FM algorithm, and the right image belongs to the SFCM algorithm. They are almost similar in terms of shape, structure, color, and size of nodes. The images generated by two algorithms, Mapper and FM, consist of two loops corresponding to 8 and 4 nodes, and the SFCM image consists of two loops with the same number of nodes. The colors of the nodes in the three images are the same, only red or blue. On the whole, the output of the SFCM algorithm is quite similar to those of the Mapper

Table 20: The parameter settings for the Two Concentric Circles dataset with Noise.

Mapper	Filter Function	Number of Intervals	Overlap Percentage	Clustering Method
	Sum	$N = 5$	$p = 5\%$	DBSCAN
FM	Filter Function	Number of Intervals	Overlap Threshold	Clustering Method
	Sum	$N = 5$	$\tau = 0.20$	DBSCAN
SFCM	Number of Clusters		Overlap Threshold	
	$N = 12$		$\tau = 0.20$	



(a) Mapper **(b)** FM **(c)** SFCM

Figure 29: The outputs for the Two Concentric Circles dataset with Noise.

and FM algorithms from the topological standpoint. Nevertheless, the essential thing in these pictures is that the data points are divided into two discrete classes with 100% accuracy for all algorithms. This result proves that the classification efficiency of the SFCM algorithm can reach equivalence in visualization with those of the Mapper and FM algorithms.

Similar to the previous experiment, the effectiveness of the SFCM algorithm is not only evaluated with the Mapper algorithm but also with the FM algorithm through the matching score and silhouette scores. All of the algorithms were run to change the overlapping parameters such that there is no change in the shape of the outputs with the other parameters fixed. Some results are recognized for percentage and thresholds as follows:

- For the Mapper algorithm, the overlapping percentage p is altered from 3% to 7%.
- For the FM algorithm, the overlapping threshold τ is altered from 0.19 to 0.40.
- For the SFCM algorithm, the overlapping threshold τ is altered from 0.15 to 0.22.

Notice that the output of the Mapper algorithm constantly forms two connected components for all values $p \leq 50\%$, and the classification has absolute accuracy. However, this parameter is only changed in the above range to detect the two circle structures.

Table 21: The evaluation results for the Two Concentric Circles dataset with Noise.

	MATCHING SCORE	SILHOUETTE SCORE
Mapper	0.821	0.208
FM	0.867	0.200
SFCM	0.900	0.306

For each algorithm, the matching and silhouette scores are also arranged the same calculation as in the previous experiment as follows:

- The matching coefficient is calculated using the k-fold cross-validation method for each case in which the overlapping parameters satisfy the non-changed shape condition. In this experiment, k is set at 10 for sub-sampling the data, the same as the sampling ratio of 0.8 used in [201]. The matching score is the mean of the matching coefficients of all cases.

- The silhouette coefficient is also calculated for each case where the overlapping parameters do not change the output shape of the algorithms. The silhouette score is the mean of the silhouette coefficients of all cases.

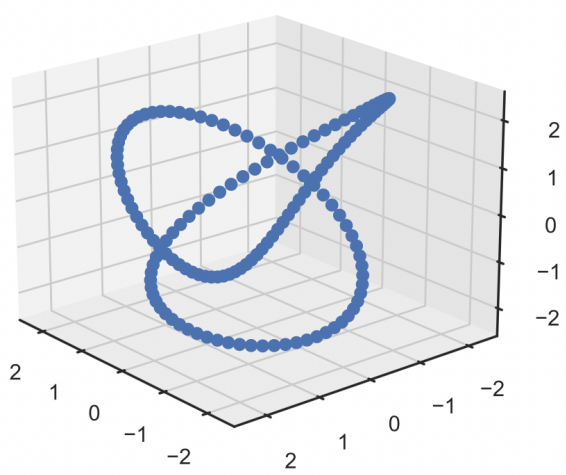
The matching and silhouette scores of the three algorithms are reported in Table 21.

4.3.3 3D Trefoil Knot Dataset

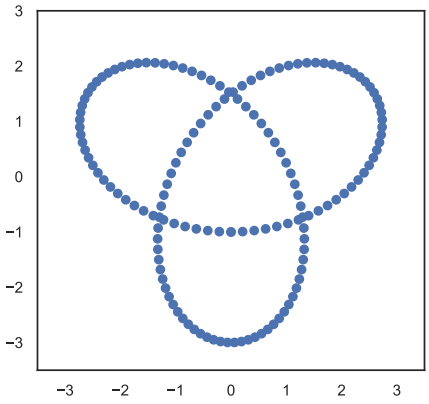
The 3D Trefoil Knot dataset has approximately 150 points that create a three-dimensional trefoil knot in Euclidean space. This dataset is used to prove the advantages of Mapper when compared to the traditional dimensionality reduction techniques, including linear (e.g., PCA) and nonlinear (e.g., t-SNE) approaches [21]. The information of this comparison can be found in [21]. The visualization of this dataset is presented in Figure 30 on the Euclidean space.

Now, three algorithms, Mapper, FM, and SFCM, process the 3D Trefoil Knot dataset. Table 22 presents the choice of parameters corresponding to each algorithm. The parameters for the Mapper algorithm are chosen so that its output can detect the circle structure as in the previous implementation [21]. The clustering algorithm used in the Mapper and Fuzzy algorithms is l^2-norm. This is a specific norm on a Euclidean vector space, strongly related to the Euclidean distance, and equals the square root of the inner product of a vector with itself. The FM and SFCM algorithms are designed so that their outputs are the same as those of the Mapper algorithm from topology. The nodes in the graphs of all algorithms are colored by the same function, the height coordinate of data points. The node's color indicates the average filter values at points in the set represented by the node. The red expresses a maximal value, and the blue says a minimal value. The node's size indicates the number of points in the set represented by the node. This size positively correlates with the number of points contained in it.

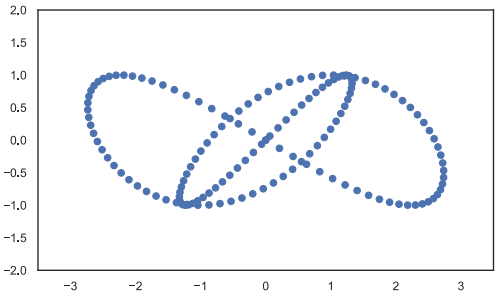
The outputs of all algorithms on this dataset are presented in Figure 31. From left to right, the images belong to the Mapper, FM, and SFCM algorithms, respectively. These images are



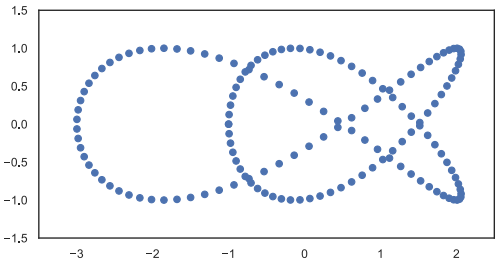
(a) $Oxyz$ space



(b) Oxy plane

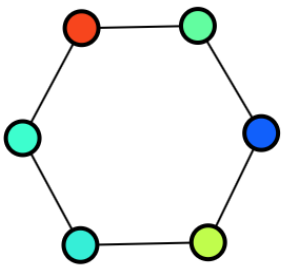


(c) Oxz plane

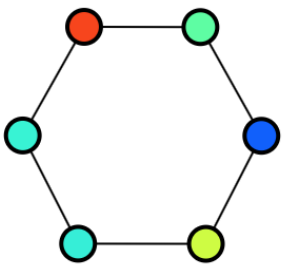


(d) Oyz plane

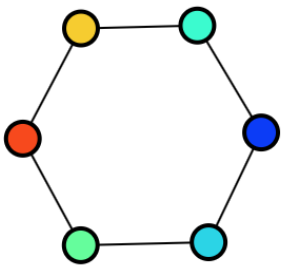
Figure 30: The visualization of the 3D Trefoil Knot dataset.



(a) Mapper



(b) FM



(c) SFCM

Figure 31: The outputs for the 3D Trefoil Knot dataset

Table 22: The parameter settings for the 3D Trefoil Knot dataset.

Mapper	FILTER FUNCTION	NUMBER OF INTERVALS	OVERLAP PERCENTAGE	CLUSTERING METHOD
	l^2-Norm	$N = 2$	$p = 50\%$	DBSCAN
FM	FILTER FUNCTION	NUMBER OF INTERVALS	OVERLAP THRESHOLD	CLUSTERING METHOD
	l^2-Norm	$N = 2$	$\tau = 0.40$	DBSCAN
SFCM	NUMBER OF CLUSTERS		OVERLAP THRESHOLD	
	$N = 6$		$\tau = 0.25$	

identical in shape, structure, and size of nodes. The color of nodes is the same for the outputs of both Mapper and FM algorithms, but it is a little different from that of the remaining algorithm. This difference is that the Mapper and FM algorithms use the filter, but the SFCM algorithm does not. Overall, the output of the SFCM algorithm is almost similar to those of the Mapper and FM algorithms concerning the topological structure.

Similarly, the effectiveness of the SFCM algorithm is evaluated with both the Mapper and FM algorithms through the matching score and silhouette scores. Once again, the overlapping parameters of all algorithms were adjusted such that there was no change in the shape of the with the other parameters fixed. Some results are recognized for percentage and thresholds as follows:

– For the Mapper algorithm, the overlapping percentage p fluctuates from 13% to 50%.

– For the FM algorithm, the overlapping threshold τ fluctuates from 0.05 to 0.42.

– For the SFCM algorithm, the overlapping threshold τ fluctuates from 0.21 to 0.30.

For each algorithm, the matching and silhouette scores are also arranged the same calculation as in the previous experiment as follows:

– The matching coefficient is calculated using the k-fold cross-validation method for each case in which the overlapping parameters satisfy the non-changed shape condition. In this experiment, k is set at 10 for sub-sampling the data, the same as the sampling ratio of 0.8 used in [201]. The matching score is the mean of the matching coefficients of all cases.

– The silhouette coefficient is also calculated for each case where the overlapping parameters do not change the output shape of the algorithms. The silhouette score is the mean of the silhouette coefficients of all cases.

The matching and silhouette scores of the three algorithms are reported in Table 23.

Table 23: The evaluation results for the 3D Trefoil Knot dataset.

	MATCHING SCORE	SILHOUETTE SCORE
Mapper	0.891	0.017
FM	0.971	0.144
SFCM	0.925	0.205

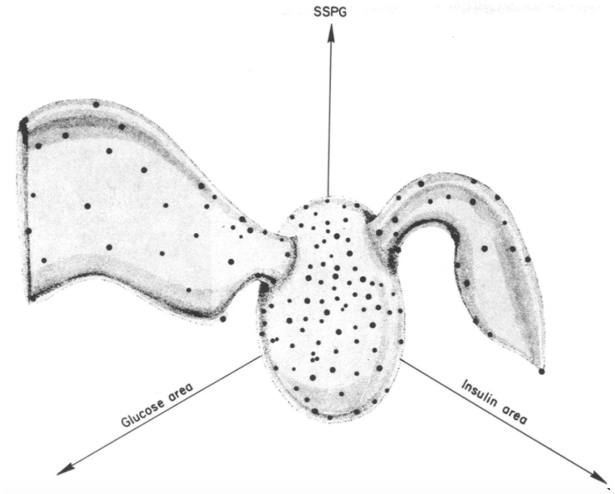


Figure 32: The three-dimensional visualization of the Reaven and Miller diabetes dataset.

4.3.4 Reaven and Miller Diabetes Dataset

The Reaven and Miller Diabetes dataset was one of the study results in the 1970s by Stanford University [181]. A total of 145 non-obese adult patients, who had diabetes and a family history of diabetes, participated in the study. For each patient, six qualitative quantities were recorded. Therefore, there were six dimensions for the Reaven and Miller Diabetes dataset, which consists of the following six features: age, relative weight, fasting plasma glucose level, test plasma glucose level, plasma insulin during the test, and steady-state plasma glucose response. In 1979, Reaven and Miller [88] visualized this dataset directly by the projection pursuit method and obtained a three-dimensional shape as shown in Figure 32. This visual representation is interpreted as a boomerang with a fat middle core and two floppy wings [88]. The authors have indicated that the central body expresses the normal patients, while the two wings outbreaking from the core express the diabetes patients suffering from different types, corresponding to the division of diabetes into the adult-onset and juvenile-onset forms.

All three algorithms, the original and improved versions, are now applied to extract topological insights from the shape of the Reaven and Miller Diabetes dataset. A detailed description of the choice of parameters in each algorithm is shown in Table 24. The parameters for the Mapper algorithm have been kept the same as the popular publications in the original paper [13]. The filter, in this case, is the KDE function in statistics. Besides, the single-linkage clustering from the scikit-learn package with the default parameter settings is used as the clustering

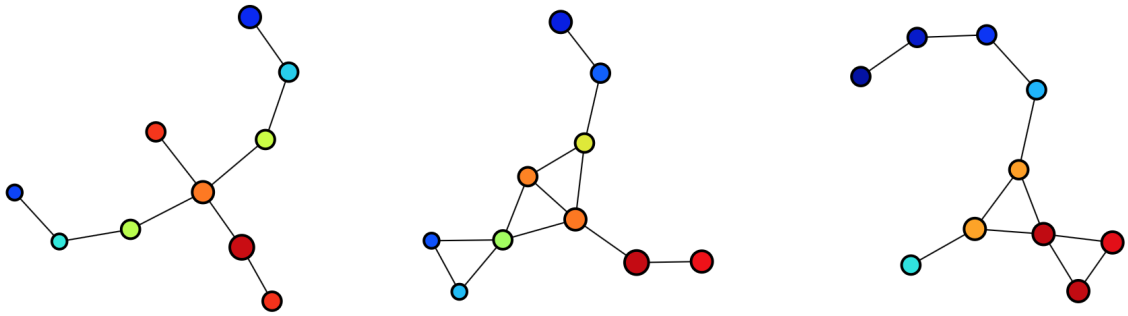
Table 24: The parameter settings for the Reaven and Miller Diabetes dataset.

Mapper	FILTER FUNCTION	NUMBER OF INTERVALS	OVERLAP PERCENTAGE	CLUSTERING METHOD
	KDE	$N = 5$	$p = 50\%$	Single-Linkage
FM	FILTER FUNCTION	NUMBER OF INTERVALS	OVERLAP THRESHOLD	CLUSTERING METHOD
	KDE	$N = 5$	$\tau = 0.06$	Single-Linkage
SFCM	NUMBER OF CLUSTERS		OVERLAP THRESHOLD	
	$N = 10$		$\tau = 0.20$	

algorithm. The FM and SFCM algorithms are exploited because their outputs were quite similar to that of the Mapper algorithm in shape and structure. The nodes in the graphs of both the algorithms were colored by the same function, which was the value of the KDE function on data points. The node's color indicates the average filter values at points in the set represented by the node. The red expresses a maximal value, and the blue says a minimal value. The node's size indicates the number of points in the set represented by the node. This size positively correlates with the number of points contained in it.

Figure 33 shows the outputs after processing by three algorithms on this diabetes dataset. Three images from left to right sequentially belong to the algorithms, Mapper, FM, and SFCM. In each algorithm, the central core that expresses the normal patients appear by red nodes. Both wings that express patients with adult-onset diabetes and juvenile-onset diabetes also appear in each algorithm by blue nodes. However, the wings in the Mapper output appear to be symmetrical. This phenomenon is not the case when the FM and SFCM algorithms are implemented. In another way, the connections between the nodes are different in the three algorithms, especially concerning the positions of the left wings. The visualizations created by the FM and SFCM algorithms are very similar to the research result of Reaven and Miller since the density of points in the two wings varies remarkably, and several points in one wing are more sparse than in those in the others. There are some triangular forms in the outputs generated by the FM and SFCM algorithms. It is caused by existent differences in covering all points of the data cloud. In the original method, the connectivity only occurs between two points to create one edge because the real-valued continuous function usually considers the filter. Nevertheless, this connectivity occurs between three points to create the triangles in the image of the new later algorithms. As well as the FM algorithm, the overlap between clusters in the SFCM algorithm cannot necessarily be pairwise and consecutive.

The process of evaluating the clustering stability through the matching score and the internal index with silhouette score is repeated for the SFCM algorithm concerning the two algorithms, Mapper and FM. Once again, all algorithms changed the overlapping parameters such



(a) Mapper (b) FM (c) SFCM

Figure 33: The outputs for the Reaven and Miller Diabetes dataset.

Table 25: The evaluation results for the Reaven and Miller Diabetes dataset.

	MATCHING SCORE	SILHOUETTE SCORE
Mapper	0.889	−0.032
FM	0.880	−0.028
SFCM	0.892	0.036

that there was no change in the output shapes in the condition that the other parameters fixed. Some results are recognized for percentage and thresholds as follows:

- For the Mapper algorithm, the overlapping percentage p is altered from 41% to 50%.
- For the FM algorithm, the overlapping threshold τ is altered from 0.056 to 0.062.
- For the SFCM algorithm, the overlapping threshold τ is altered from 0.20 to 0.24

For each algorithm, the matching score and silhouette score are also carried out the same computation as in the previous experiments as follows:

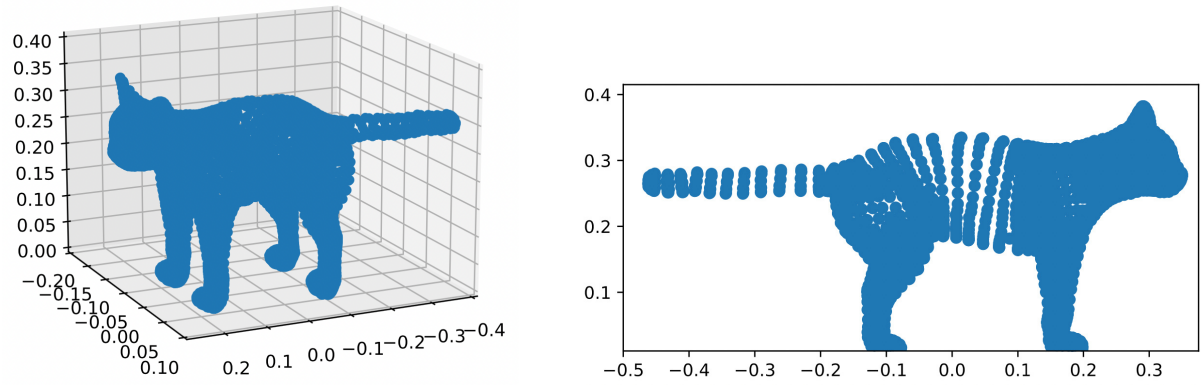
- The matching coefficient is calculated using the k-fold cross-validation method for each case in which the overlapping parameters satisfy the non-changed shape condition. In this experiment, k is set at 10 for sub-sampling the data, the same as the sampling ratio of 0.8 used in [201]. The matching score is the mean of the matching coefficients of all cases.
- The silhouette coefficient is also calculated for each case where the overlapping parameters do not change the output shape of the algorithms. The silhouette score is the mean of the silhouette coefficients of all cases.

The matching and silhouette scores of the three algorithms are reported in Table 25.

Overall, for the four real-world datasets, the experiments focus on the output visualization from the topological standpoint, the clustering stability through the matching score, and the

Table 26: The parameter settings and run-time report for the experiments on the large high-dimensional datasets.

DATASET	DIMENSION	NO. CLUSTERS	OVERLAPPING THRESHOLD	RUN-TIME
3D Lion	$(5000; 3)$	12	0.180	0.65
3D Cat	$(7206; 3)$	10	0.217	0.52
3D Horse	$(8430; 3)$	20	0.117	5.01
3D Road	$(434874; 4)$	12	0.150	119.76
Covertype	$(581012; 54)$	12	0.160	752.94



(a) $Oxyz$ space **(b)** Oxz plane

Figure 34: The three-dimensional visualization of the Cat dataset.

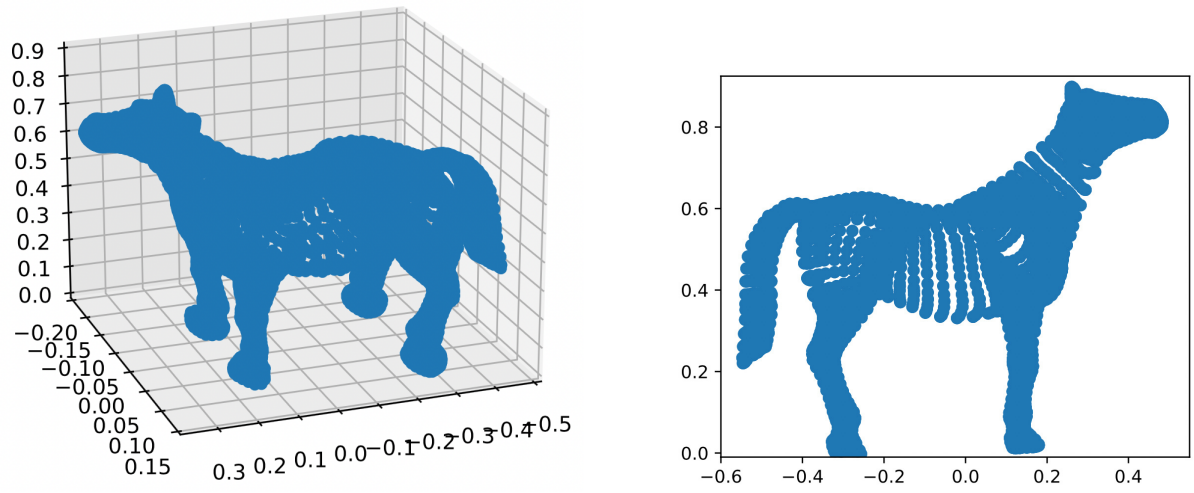
internal index with silhouette score for the three algorithms. All three algorithms, Mapper, FM, and SFCM, have been thoroughly implemented and achieved positive results as follows:

(1) Firstly, the results generated by the SFCM algorithm are pretty similar to those of the Mapper and FM algorithms on the topological standpoint in cases where the respective parameters are well-chosen.

(2) Secondly, the clustering stability based on the matching score and the internal index based on the silhouette score of the SFCM algorithm is better than those of the Mapper and FM algorithm in most experimental cases.

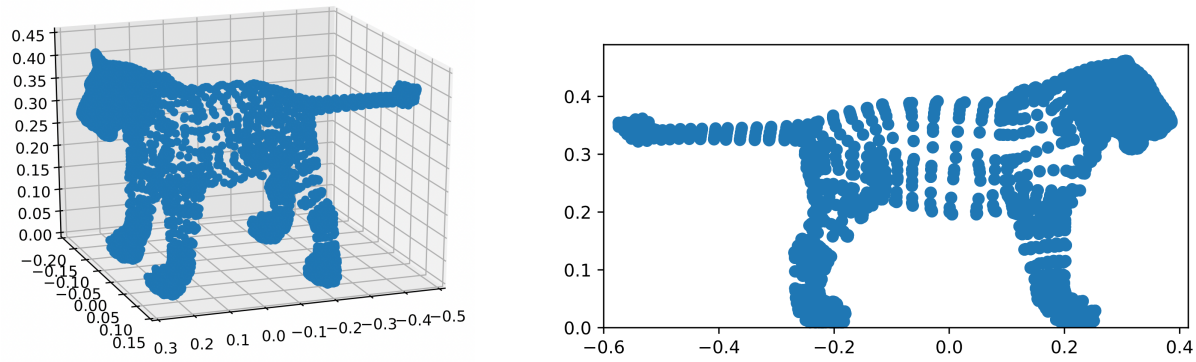
To conclude this section, the SFCM algorithm is applied to visualize some large multi-dimensional datasets, including 3D Cat, 3D Horse, 3D Lion, 3D Road Network, and Covertype, to prove the algorithm's capability on big data. The 3D datasets of Lion, Cat, and Horse were taken from the examples in the open-source codebase of the KeplerMapper [112, 113]. The 3D Road Network dataset [205] and the Covertype dataset [206] were taken from the UCI Machine Learning Repository.

The choice of the parameters and the running time in seconds when using the SFCM algorithm for each dataset are reported in Table 26. The three-dimensional visualizations of the



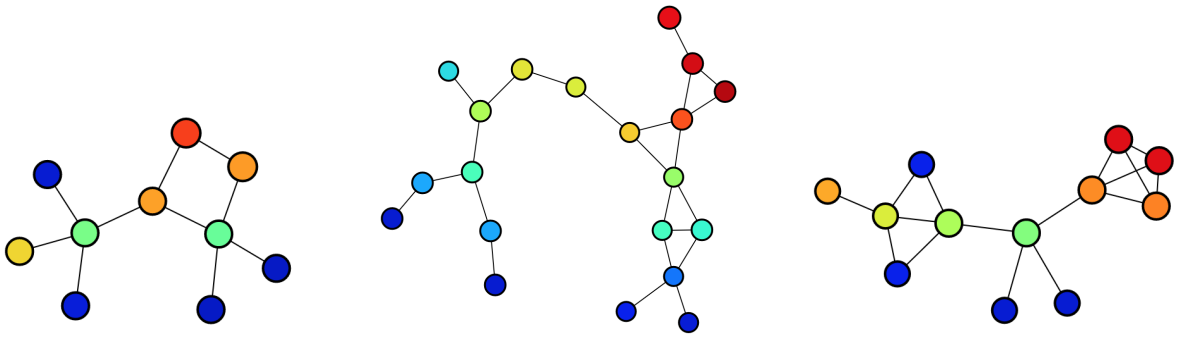
(a) $Oxyz$ space **(b)** Oxz plane

Figure 35: The three-dimensional visualization of the Horse dataset.



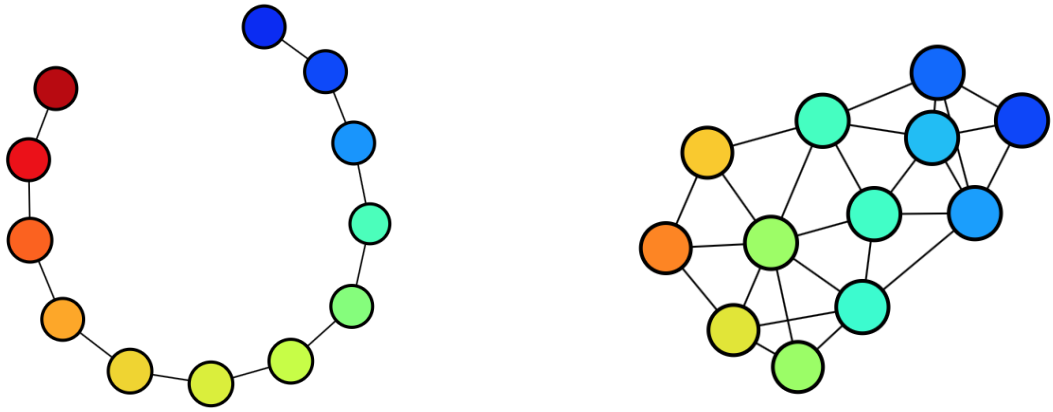
(a) $Oxyz$ space **(b)** Oxz plane

Figure 36: The three-dimensional visualization of the Lion dataset.



(a) Cat **(b)** Horse **(c)** Lion

Figure 37: The outputs for the animal datasets.



(a) 3D Road Network (b) Covertype

Figure 38: The outputs for the 3D Road Network and Covertype datasets.

animal datasets are presented in Figures 34, 35, 36. The results of using the SFCM algorithm to mine these three-dimensional datasets are shown in Figures 37 and 38

The run-time results are perfectly acceptable for the high-dimensional datasets. Note that the run-time between the new proposed algorithm and the original algorithms is not comparable because their run-time result depends heavily on the time complexity of the filter itself. The results are rather impressive for the 3D animal datasets when the height coordinate of data points colors nodes of the graph. The red expresses maximum value, and blue expresses minimum value. The color-changing from red to blue indicates the corresponding values that vary from high to low. Some parts of the animal bodies are shown clearly on the graph: blue nodes present the feet and red nodes offer the heads.

4.4 Discussion

In this chapter, the SFCM algorithm is proposed as a fuzzy clustering algorithm endowed with the ability to create a simple, practical, intuitive topological summary for high-dimensional datasets. It has been generated from the FCM algorithm by carefully combining the unique power for shape detection of the Mapper algorithm. On the one hand, the FCM algorithm has outstanding features in simplifying and visualizing data with qualitative analysis. On the other hand, the SFCM algorithm also helps the Mapper algorithm simplify the selection of parameters to obtain the most informative presentation. Moreover, covering the data space by clusters created by the FCM algorithm is both a breakthrough and a logical idea with high feasibility.

Experiments on the four real-world datasets demonstrate the effectiveness of the SFCM algorithm. From the topological standpoint, this method can produce results similar to those of the previous popular algorithms, Mapper and FM. Besides, the matching score, which presents the clustering stability, and the silhouette score, which offers the internal index of the novel proposed algorithm, are better than those of the Mapper and FM in most experimental cases.

In a certain sense, the SFCM algorithm is considered a particular enhanced case of the FM algorithm. The overlapping threshold for this specific version, like the FM algorithm, is chosen based on the positive results. How to optimize this parameter to an optimal value is still an interesting question. Although the optimization capabilities of this algorithm are thoroughly proven in the experiments, its practical applicability needs to be further examined. In addition, the theoretical framework of this method needs to be developed towards in-depth analysis by using the persistent homology theory. The study of the SFCM algorithm in cases when the output is a general simplicial complex also requires more attention. Last but not least, the improvements to both the two-component algorithms should be updated to take account of this work. These problems are expected to be addressed thoroughly soon to improve this method regarding both theory and applications.

5 Conclusion

In Chapter 5, the summary of our completed works according to the initial aims is briefly presented through the author's main contributions in publishing activities. In addition, the future research directions are also discussed and planned to serve as a guideline for the author after finishing the doctoral course. Therefore, the rest of this chapter is organized into two sections corresponding to the two contents just mentioned, including contributions in Section 5.1 and directions in Section 5.2.

5.1 Contribution

The first goal of the thesis is to summarize the theoretical foundations and practical applications of the Mapper algorithm in the flow of literature with improved versions and various implementations. This goal is achieved in Chapter 2 of this thesis when this intelligent algorithm is reviewed with specific descriptions and intuitive, easy-to-understand illustrations. Its variations and applications are also presented over time systematically and thoroughly. At the same time, the popular available packages that activate this algorithm as the core of their operations are also briefly introduced. Furthermore, its current limitations are also discussed to guide future research and development. This chapter has been aggregated into a manuscript for publication as a peer-reviewed scientific paper in a reputable journal [P7].

The second goal of the thesis is to optimize the cover choice of the Mapper algorithm in the direction of dividing the filter range automatically into irregular intervals with a random overlapping percentage by using the FCM algorithm. This goal is achieved in Chapter 3 of this thesis when a novel algorithm, named Fuzzy Mapper, is proposed on the foundation of the Mapper algorithm to solve the problem of automating in dividing cover intervals with an arbitrary percentage of overlap. The experimental results are analyzed and compared with those of the original method, the Mapper algorithm, through the output visualization and the silhouette coefficient score in the clustering evaluation. This chapter was published as a peer-reviewed scientific paper in a reputable journal, Knowledge-Based Systems [P1].

The third goal of the thesis is to propose a novel method for mining data that can exhibit the same clustering ability as the Fuzzy C-Means algorithm and reveal some meaningful relationships by visualizing the global data shape supplied by the Mapper algorithm. This goal is achieved in Chapter 4 of this thesis when another novel algorithm, named Shape Fuzzy C-Means, is proposed by combining the advantages of the Fuzzy C-Means algorithm with outstanding features of the Mapper algorithm. The algorithm performance is demonstrated through a comparative analysis involving the original algorithm, Mapper, and the other fuzzy set-based improved algorithm, Fuzzy Mapper. The comparison is conducted concerning output visualization in the topological sense and cluster. This chapter was published as a peer-reviewed scientific paper in a reputable journal, IEEE Transactions on Fuzzy Systems [P3]

5.2 Orientation

Although the thesis has achieved the set goals, there are still many challenges. Topological Data Analysis is already a powerful emerging field and is expanding without limits to the very vibrant areas of artificial intelligence such as machine learning and deep learning. This development is not only a challenge, but also an opportunity when researching this field. In the coming period, we will focus on the following research directions:

(1) Evaluating the complexity and overcoming the limitations of our algorithms to improve and develop them towards using the superior features of fuzzy clustering as well as variants of set theory such as rough set, vague set, soft set, and neutrosophic set [207].

(2) Developing method(s) to evaluate Mapper-type algorithms in performance and stability [131, 208].

(3) Studying the characteristics of the Mapper algorithm based on theoretical framework in close relationship with Reeb Graph and Discrete Morse Theory [209].

(4) Improving the ability to simplify complex high-dimensional data but still ensure the structural features of the Mapper algorithm on complex networks [210, 211, 70] for mining and learning purposes.

We hope these directions will bring positive research signals soon. The initial achievements during my time as a doctoral student reported in this thesis are an excellent impetus for the next steady steps to implement new ideas on the research way of Topological Data Analysis.

References

1. SNÁŠEL, Václav; NOWAKOVÁ, Jana; XHAFA, Fatos; BAROLLI, Leonard. Geometrical and Topological Approaches to Big Data. *Future Generation Computer Systems*. 2017, vol. 67, pp. 286–296. ISSN 0167-739X. Available from DOI: 10.1016/j.future.2016.06.005.
2. CARLSSON, Gunnar. Topology and Data. *Bulletin of the American Mathematical Society*. 2009, vol. 46, no. 2, pp. 255–308. ISSN 02730979. Available from DOI: 10.1090/S0273-0979-09-01249-X.
3. CARLSSON, Gunnar. Topological Pattern Recognition for Point Cloud Data. *Acta Numerica*. 2014, vol. 23, pp. 289–368. Available from DOI: 10.1017/S0962492914000051.
4. OTTER, Nina; PORTER, Mason A.; TILLMANN, Ulrike; GRINDROD, Peter; HARRINGTON, Heather A. A Roadmap for the Computation of Persistent Homology. *EPJ Data Science*. 2017, vol. 6, no. 1, p. 17. ISSN 21931127. Available from DOI: 10.1140/epjds/s13688-017-0109-5.
5. PATANIA, Alice; VACCARINO, Francesco; PETRI, Giovanni. Topological Analysis of Data. *EPJ Data Science*. 2017, vol. 6, no. 1, p. 7. ISSN 2193-1127. Available from DOI: 10.1140/epjds/s13688-017-0104-x.
6. BOISSONNAT, Jean-Daniel; CHAZAL, Frédéric; YVINEC, Mariette. *Geometric and Topological Inference*. Cambridge University Press, 2018. Cambridge Texts in Applied Mathematics. ISBN 9781108410892. Available from DOI: 10.1017/9781108297806.
7. ZOMORODIAN, Afra. Topological Data Analysis. *Advances in Applied and Computational Topology*. 2012, vol. 70, pp. 1–39.
8. BEYER, David. Gurjeet Singh: Using Topology to Uncover the Shape of Your Data. In: *The Future of Machine Intelligence: Perspectives from Leading Practitioners*. O'Reilly Media, 2016, chap. 7. ISBN 9781491932285. Available also from: https://www.oreilly.com/data/free/files/future-of-machine-intelligence.pdf.
9. GHRIST, Robert. Barcodes: The Persistent Topology of Data. *Bulletin of the American Mathematical Society*. 2008, vol. 45, no. 1, pp. 61–75. Available from DOI: 10.1090/S0273-0979-07-01191-3.
10. WASSERMAN, Larry. Topological Data Analysis. *Annual Review of Statistics and Its Application*. 2018, vol. 5, no. 1, pp. 501–532. Available from DOI: 10.1146/annurev-statistics-031017-100045.
11. ZOMORODIAN, Afra J. *Topology for Computing*. Cambridge University Press, 2005. Cambridge Monographs on Applied and Computational Mathematics. ISBN 9781139442633. Available from DOI: 10.1017/CBO9780511546945.

12. EDELSBRUNNER, Herbert; HARER, John. Persistent Homology - A Survey. *Contemporary mathematics*. 2008, vol. 453, pp. 257–282. Available from DOI: 10.1090/conm/453/08802.

13. SINGH, Gurjeet; MEMOLI, Facundo; CARLSSON, Gunnar. Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. In: BOTSCH, M.; PAJAROLA, R.; CHEN, B.; ZWICKER, M. (eds.). *Eurographics Symposium on Point-Based Graphics*. The Eurographics Association, 2007. Available from DOI: 10.2312/SPBG/SPBG07/091-100.

14. CARLSSON, Gunnar. Topological Methods for Data Modelling. *Nature Reviews Physics*. 2020, vol. 2, no. 12, pp. 697–708. ISSN 2522-5820. Available from DOI: 10.1038/s42254-020-00249-3.

15. SCOVILLE, Nicholas A. *Discrete Morse Theory*. American Mathematical Society, 2019. Student Mathematical Library. ISBN 9781470452988. Available also from: https://books.google.cz/books?id=hduyDwAAQBAJ.

16. KNUDSON, Kevin P. Discrete Morse Theory by Nicholas Scoville. *The American Mathematical Monthly*. 2020, vol. 127, no. 8, pp. 763–768. Available from DOI: 10.1080/00029890.2020.1792244.

17. LUM, P. Y.; SINGH, G.; LEHMAN, A.; ISHKANOV, T.; VEJDEMO-JOHANSSON, M.; ALAGAPPAN, M.; CARLSSON, J.; CARLSSON, G. Extracting Insights From the Shape of Complex Data Using Topology. *Scientific Reports*. 2013, vol. 3, p. 1236. ISSN 20452322. Available from DOI: 10.1038/srep01236.

18. RABADAN, Raul; BLUMBERG, Andrew J. *Topological Data Analysis for Genomics and Evolution: Topology in Biology*. Cambridge University Press, 2019. ISBN 9781107159549. Available from DOI: 10.1017/9781316671665.

19. SALCH, Andrew; REGALSKI, Adam; ABDALLAH, Hassan; SURYADEVARA, Raviteja; CATANZARO, Michael J.; DIWADKAR, Vaibhav A. From Mathematics to Medicine: a Practical Primer on Topological Data Analysis (TDA) and the Development of Related Analytic Tools for the Functional Discovery of Latent Structure in fMRI Data. *PLOS ONE*. 2021, vol. 16, no. 8, pp. 1–33. Available from DOI: 10.1371/journal.pone.0255859.

20. SAGGAR, Manish; SPORNS, Olaf; GONZALEZ-CASTILLO, Javier; BANDETTINI, Peter A.; CARLSSON, Gunnar; GLOVER, Gary; REISS, Allan L. Towards a New Approach to Reveal Dynamical Organization of the Brain Using Topological Data Analysis. *Nature Communications*. 2018, vol. 9, no. 1, p. 1399. ISSN 20411723. Available from DOI: 10.1038/s41467-018-03664-4.

21. GENIESSE, Caleb; SPORNS, Olaf; PETRI, Giovanni; SAGGAR, Manish. Generating Dynamical Neuroimaging Spatiotemporal Representations (DyNeuSR) using Topological Data Analysis. *Network Neuroscience*. 2019, vol. 3, no. 3, pp. 763–778. ISSN 2472-1751. Available from DOI: 10.1162/netn_a_00093.

22. SIZEMORE, Ann E.; PHILLIPS-CREMINS, Jennifer E.; GHRIST, Robert; BASSETT, Danielle S. The Importance of the Whole: Topological Data Analysis for the Network Neuroscientist. *Network Neuroscience*. 2019, vol. 3, no. 3, pp. 656–673. ISSN 2472-1751. Available from DOI: 10.1162/netn_a_00073.

23. SAGGAR, Manish; SHINE, James M.; LIÉGEOIS, Raphaël; DOSENBACH, Nico U. F.; FAIR, Damien. Precision Dynamical Mapping Using Topological Data Analysis Reveals a Unique Hub-like Transition State at Rest. *bioRxiv*. 2021. Available from DOI: 10.1101/2021.08.05.455149.

24. CÁMARA, Pablo G. Topological methods for genomics: Present and future directions. *Current Opinion in Systems Biology*. 2017, vol. 1, pp. 95–101. ISSN 2452-3100. Available from DOI: 10.1016/j.coisb.2016.12.007.

25. LEE, Jin-Ku; LIU, Zhaoqi; SA, Jason K; SHIN, Sang; WANG, Jiguang; BORDYUH, Mykola; CHO, Hee Jin; ELLIOTT, Oliver; CHU, Timothy; CHOI, Seung Won; ROSENBLOOM, Daniel I S; LEE, In-Hee; SHIN, Yong Jae; KANG, Hyun Ju; KIM, Donggeon; KIM, Sun Young; SIM, Moon-Hee; KIM, Jusun; LEE, Taehyang; SEO, Yun Jee; SHIN, Hyemi; LEE, Mijeong; KIM, Sung Heon; KWON, Yong-Jun; OH, Jeong-Woo; SONG, Minsuk; KIM, Misuk; KONG, Doo-Sik; CHOI, Jung Won; SEOL, Ho Jun; LEE, Jung-Il; KIM, Seung Tae; PARK, Joon Oh; KIM, Kyoung-Mee; SONG, Sang-Yong; LEE, Jeong-Won; KIM, Hee-Cheol; LEE, Jeong Eon; CHOI, Min Gew; SEO, Sung Wook; SHIM, Young Mog; ZO, Jae Ill; JEONG, Byong Chang; YOON, Yeup; RYU, Gyu Ha; KIM, Nayoung K D; BAE, Joon Seol; PARK, Woong-Yang; LEE, Jeongwu; VERHAAK, Roel G W; IAVARONE, Antonio; LEE, Jeeyun; RABADAN, Raul; NAM, Do-Hyun. Pharmacogenomic Landscape of Patient-derived Tumor Cells Informs Precision Oncology Therapy. *Nature Genetics*. 2018, vol. 50, no. 10, pp. 1399–1411. ISSN 1546-1718. Available from DOI: 10.1038/s41588-018-0209-6.

26. RABADÁN, Raúl; MOHAMEDI, Yamina; RUBIN, Udi; CHU, Tim; ALGHALITH, Adam N; ELLIOTT, Oliver; ARNÉS, Luis; CAL, Santiago; OBAYA, Álvaro J; LEVINE, Arnold J; CÁMARA, Pablo G. Identification of Relevant Genetic Alterations in Cancer Using Topological Data Analysis. *Nature Communications*. 2020, vol. 11, no. 1, p. 3808. ISSN 2041-1723. Available from DOI: 10.1038/s41467-020-17659-7.

27. ROSENBERG, Ayelet; SAGGAR, Manish; ROGU, Peter; LIMOGES, Aaron W.; SANDI, Carmen; MOSHAROV, Eugene V.; DUMITRIU, Dani; ANACKER, Christoph; PICARD, Martin. Mouse Brain-wide Mitochondrial Connectivity Anchored in Gene, Brain and Behavior. *bioRxiv*. 2021. Available from DOI: 10.1101/2021.06.02.446767.

28. CARLSSON, Gunnar; GABRIELSSON, Rickard Brüel. Topological Approaches to Deep Learning. In: BAAS, Nils A.; CARLSSON, Gunnar E.; QUICK, Gereon; SZYMIK, Markus; THAULE, Marius (eds.). *Topological Data Analysis*. Cham: Springer International Publishing, 2020, pp. 119–146. ISBN 978-3-030-43408-3. Available from DOI: 10.1007/978-3-030-43408-3_5.

29. CÔTÉ-ALLARD, Ulysse; CAMPBELL, Evan; PHINYOMARK, Angkoon; LAVIOLETTE, François; GOSSELIN, Benoit; SCHEME, Erik. Interpreting Deep Learning Features for Myoelectric Control: A Comparison With Handcrafted Features. *Frontiers in Bioengineering and Biotechnology*. 2020, vol. 8, p. 158. ISSN 2296-4185. Available from DOI: 10.3389/fbioe.2020.00158.

30. HATCHER, Allen. *Algebraic Topology*. Cambridge University Press, 2002. ISBN 9780521795401. Available also from: https://pi.math.cornell.edu/~hatcher/AT/AT.pdf.

31. MUNKRES, James R. Munkres. *Elements Of Algebraic Topology*. CRC Press, 2018. ISBN 9780429962462. Available from DOI: 10.1201/9780429493911.

32. EDELSBRUNNER, Herbert; HARER, John L. *Computational Topology: An Introduction*. American Mathematical Society, 2010. Applied Mathematics. ISBN 9780821849255. Available from DOI: 10.1090/mbk/069.

33. HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer New York, 2009. Springer Series in Statistics. ISBN 9780387848587. Available from DOI: 10.1007/978-0-387-84858-7.

34. HAN, Jiawei; KAMBER, Micheline; PEI, Jian. *Data Mining: Concepts and Techniques*. Elsevier Science, 2011. The Morgan Kaufmann Series in Data Management Systems. ISBN 9780123814807. Available from DOI: 10.1016/C2009-0-61819-5.

35. BUI, Quang-Thinh; VO, Bay; DO, Hoang-Anh Nguyen; HUNG, Nguyen Quoc Viet; SNASEL, Vaclav. F-Mapper: A Fuzzy Mapper Clustering Algorithm. *Knowledge-Based Systems*. 2020, vol. 189, p. 105107. ISSN 0950-7051. Available from DOI: 10.1016/j.knosys.2019.105107.

36. CHAZAL, Frédéric; MICHEL, Bertrand. An Introduction to Topological Data Analysis: Fundamental and Practical Aspects for Data Scientists. *Frontiers in Artificial Intelligence*. 2021, vol. 4, p. 108. ISSN 2624-8212. Available from DOI: 10.3389/frai.2021.667963.

37. BUI, Quang-Thinh; VO, Bay; SNASEL, Vaclav; PEDRYCZ, Witold; HONG, Tzung-Pei; NGUYEN, Ngoc-Thanh; CHEN, Mu-Yen. SFCM: A Fuzzy Clustering Algorithm of Extracting the Shape Information of Data. *IEEE Transactions on Fuzzy Systems*. 2021, vol. 29, no. 1, pp. 75–89. Available from DOI: 10.1109/TFUZZ.2020.3014662.

38. ZADEH, L.A. Fuzzy Sets. *Information and Control*. 1965, vol. 8, no. 3, pp. 338–353. ISSN 0019-9958. Available from DOI: 10.1016/S0019-9958(65)90241-X.

39. MENG, Fanyong; TANG, Jie; FUJITA, Hamido. Linguistic Intuitionistic Fuzzy Preference Relations and Their Application to Multi-criteria Decision Making. *Information Fusion*. 2019, vol. 46, pp. 77–90. ISSN 1566-2535. Available from DOI: 10.1016/j.inffus.2018.05.001.

40. HERRERA-VIEDMA, Enrique; PALOMARES, Iván; LI, Cong-Cong; CABRERIZO, Francisco Javier; DONG, Yucheng; CHICLANA, Francisco; HERRERA, Francisco. Revisiting Fuzzy and Linguistic Decision Making: Scenarios and Challenges for Making Wiser Decisions in a Better Way. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2021, vol. 51, no. 1, pp. 191–208. Available from DOI: 10.1109/TSMC.2020.3043016.

41. FENG, Feng; FUJITA, Hamido; ALI, Muhammad Irfan; YAGER, Ronald R.; LIU, Xiaoyan. Another View on Generalized Intuitionistic Fuzzy Soft Sets and Related Multiattribute Decision Making Methods. *IEEE Transactions on Fuzzy Systems*. 2019, vol. 27, no. 3, pp. 474–488. Available from DOI: 10.1109/TFUZZ.2018.2860967.

42. ZHA, Quanbo; DONG, Yucheng; ZHANG, Hengjie; CHICLANA, Francisco; HERRERA-VIEDMA, Enrique. A Personalized Feedback Mechanism Based on Bounded Confidence Learning to Support Consensus Reaching in Group Decision Making. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2021, vol. 51, no. 6, pp. 3900–3910. Available from DOI: 10.1109/TSMC.2019.2945922.

43. KRISHNAPURAM, Raghu; JOSHI, Anupam; NASRAOUI, Olfa; YI, Liyu. Low-complexity Fuzzy Relational Clustering Algorithms for Web Mining. *IEEE Transactions on Fuzzy Systems*. 2001, vol. 9, no. 4, pp. 595–607. Available from DOI: 10.1109/91.940971.

44. LIN, Chun-Wei; HONG, Tzung-Pei. A Survey of Fuzzy Web Mining. *WIREs Data Mining and Knowledge Discovery*. 2013, vol. 3, no. 3, pp. 190–199. Available from DOI: 10.1002/widm.1091.

45. QIU, Shuo; WANG, Boyang; LI, Ming; LIU, Jiqiang; SHI, Yanfeng. Toward Practical Privacy-Preserving Frequent Itemset Mining on Encrypted Cloud Data. *IEEE Transactions on Cloud Computing*. 2020, vol. 8, no. 1, pp. 312–323. Available from DOI: 10.1109/TCC.2017.2739146.

46. HALIM, Zahid; ALI, Omer; GHUFRAN KHAN, Muhammad. On the Efficient Representation of Datasets as Graphs to Mine Maximal Frequent Itemsets. *IEEE Transactions on Knowledge and Data Engineering*. 2021, vol. 33, no. 4, pp. 1674–1691. Available from DOI: 10.1109/TKDE.2019.2945573.

47. ROY, Shaswati; MAJI, Pradipta. Medical Image Segmentation by Partitioning Spatially Constrained Fuzzy Approximation Spaces. *IEEE Transactions on Fuzzy Systems*. 2020, vol. 28, no. 5, pp. 965–977. Available from DOI: 10.1109/TFUZZ.2020.2965896.

48. MAJI, Pradipta; MAHAPATRA, Suman. Circular Clustering in Fuzzy Approximation Spaces for Color Normalization of Histological Images. *IEEE Transactions on Medical Imaging*. 2020, vol. 39, no. 5, pp. 1735–1745. Available from DOI: 10.1109/TMI.2019.2956944.

49. AMIRKHANI, Abdollah; KOLAHDOOZI, Mojtaba; WANG, Chen; KURGAN, Lukasz A. Prediction of DNA-Binding Residues in Local Segments of Protein Sequences with Fuzzy Cognitive Maps. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. 2020, vol. 17, no. 4, pp. 1372–1382. Available from DOI: 10.1109/TCBB.2018.2890261.

50. BEZDEK, James C.; EHRLICH, Robert; FULL, William. FCM: The Fuzzy C-Means Clustering Algorithm. *Computers & Geosciences*. 1984, vol. 10, no. 2, pp. 191–203. ISSN 0098-3004. Available from DOI: 10.1016/0098-3004(84)90020-7.

51. DUNN, J. C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-separated Clusters. *Journal of Cybernetics*. 1973, vol. 3, no. 3, pp. 32–57. Available from DOI: 10.1080/01969727308546046.

52. YU, Jian; CHENG, Qiansheng; HUANG, Houkuan. Analysis of The Weighting Exponent in the FCM. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*. 2004, vol. 34, no. 1, pp. 634–639. Available from DOI: 10.1109/TSMCB.2003.810951.

53. PAL, N.R.; BEZDEK, J.C. On Cluster Validity for the Fuzzy C-Means Model. *IEEE Transactions on Fuzzy Systems*. 1995, vol. 3, no. 3, pp. 370–379. Available from DOI: 10.1109/91.413225.

54. PEDRYCZ, Witold; VALENTE DE OLIVEIRA, JosÉ. A Development of Fuzzy Encoding and Decoding Through Fuzzy Clustering. *IEEE Transactions on Instrumentation and Measurement*. 2008, vol. 57, no. 4, pp. 829–837. Available from DOI: 10.1109/TIM.2007.913809.

55. CHEN, Yiran; VOLIĆ, Ismar. Topological data analysis model for the spread of the coronavirus. *PLOS ONE*. 2021, vol. 16, no. 8, pp. 1–24. Available from DOI: 10.1371/journal.pone.0255584.

56. YAO, Yuan; SUN, Jian; HUANG, Xuhui; BOWMAN, Gregory R.; SINGH, Gurjeet; LESNICK, Michael; GUIBAS, Leonidas J.; PANDE, Vijay S.; CARLSSON, Gunnar. Topological Methods for Exploring Low-density States in Biomolecular Folding Pathways. *The Journal of Chemical Physics*. 2009, vol. 130, no. 14, p. 144115. Available from DOI: 10.1063/1.3103496.

57. LI, Li; CHENG, Wei-Yi; GLICKSBERG, Benjamin S.; GOTTESMAN, Omri; TAMLER, Ronald; CHEN, Rong; BOTTINGER, Erwin P.; DUDLEY, Joel T. Identification of Type 2 Diabetes Subgroups Through Topological Analysis of Patient Similarity. *Science Translational Medicine*. 2015, vol. 7, no. 311, 311ra174. Available from DOI: 10.1126/scitranslmed.aaa9364.

58. NIELSON, Jessica L.; PAQUETTE, Jesse; LIU, Aiwen W.; GUANDIQUE, Cristian F.; TOVAR, C. Amy; INOUE, Tomoo; IRVINE, Karen Amanda; GENSEL, John C.; KLOKE, Jennifer; PETROSSIAN, Tanya C.; LUM, Pek Y.; CARLSSON, Gunnar E.; MANLEY, Geoffrey T.; YOUNG, Wise; BEATTIE, Michael S.; BRESNAHAN, Jacqueline C.; FERGUSON, Adam R. Topological Data Analysis for Discovery in Preclinical Spinal Cord Injury and Traumatic Brain Injury. *Nature Communications*. 2015, vol. 6, p. 8581. ISSN 20411723. Available from DOI: 10.1038/ncomms9581.

59. BRUNO, Jennifer Lynn; HOSSEINI, S. M. Hadi; SAGGAR, Manish; QUINTIN, Eve-Marie; RAMAN, Mira Michelle; REISS, Allan L. Altered Brain Network Segregation in Fragile X Syndrome Revealed by Structural Connectomics. *Cerebral Cortex*. 2016, vol. 27, no. 3, pp. 2249–2259. ISSN 1047-3211. Available from DOI: 10.1093/cercor/bhw055.

60. ROSSI-DEVRIES, Jasmine; PEDOIA, Valentina; SAMAAN, Michael A.; FERGUSON, Adam R.; SOUZA, Richard B.; MAJUMDAR, Sharmila. Using Multidimensional Topological Data Analysis to Identify Traits of Hip Osteoarthritis. *Journal of Magnetic Resonance Imaging*. 2018, vol. 48, no. 4, pp. 1046–1058. Available from DOI: 10.1002/jmri.26029.

61. FOURNIER, Margot; SCOLAMIERO, Martina; GHOLAM-REZAEE, Mehdi M; CLEUSIX, Martine; JENNI, Raoul; FERRARI, Carina; GOLAY, Philippe; BAUMANN, Philipp S; CUENOD, Michel; CONUS, Philippe; DO, Kim Q; HESS, Kathryn. Topology Predicts Long-term Functional Outcome in Early Psychosis. *Molecular Psychiatry*. 2020. ISSN 1476-5578. Available from DOI: 10.1038/s41380-020-0826-1.

62. CARR, Ewan; CARRIÈRE, Mathieu; MICHEL, Bertrand; CHAZAL, Frédéric; INIESTA, Raquel. Identifying Homogeneous Subgroups of Patients and Important Features: A Topological Machine Learning Approach. *BMC Bioinformatics*. 2021, vol. 22, no. 1, p. 449. ISSN 1471-2105. Available from DOI: 10.1186/s12859-021-04360-9.

63. CAMPBELL, Evan; PHINYOMARK, Angkoon; SCHEME, Erik. Feature Extraction and Selection for Pain Recognition Using Peripheral Physiological Signals. *Frontiers in Neuroscience*. 2019, vol. 13, p. 437. ISSN 1662-453X. Available from DOI: 10.3389/fnins.2019.00437.

64. DEY, Tamal K.; MÉMOLI, Facundo; WANG, Yusu. Multiscale Mapper: Topological Summarization via Codomain Covers. In: *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*. 2016, vol. 2, pp. 997–1013. ISBN 9781510819672. Available from DOI: 10.1137/1.9781611974331.ch71.

65. DEY, Tamal K.; MÉMOLI, Facundo; WANG, Yusu. Topological Analysis of Nerves, Reeb Spaces, Mappers, and Multiscale Mappers. In: ARONOV, Boris; KATZ, Matthew J. (eds.). *33rd International Symposium on Computational Geometry (SoCG 2017)*. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017, vol. 77, 36:1–36:16. Leibniz In-

ternational Proceedings in Informatics (LIPIcs). ISBN 978-3-95977-038-5. ISSN 1868-8969. Available from DOI: 10.4230/LIPIcs.SoCG.2017.36.

66. HAJIJ, Mustafa; ASSIRI, Basem; ROSEN, Paul. Parallel Mapper. In: ARAI, Kohei; KAPOOR, Supriya; BHATIA, Rahul (eds.). *Proceedings of the Future Technologies Conference (FTC) 2020, Volume 2*. Cham: Springer International Publishing, 2021, pp. 717–731. ISBN 978-3-030-63089-8. Available from DOI: 10.1007/978-3-030-63089-8_47.

67. HAJIJ, Mustafa; ROSEN, Paul; WANG, Bei. *Mapper on Graphs for Network Visualization*. 2019. Available from arXiv: 1804.11242.

68. DŁOTKO, Paweł. *Ball Mapper: A Shape Summary for Topological Data Analysis*. 2019. Available from arXiv: 1901.07410.

69. KANG, Sung Jin; LIM, Yaeji. Ensemble Mapper. *Stat*. 2021, vol. 10, no. 1, e405. Available from DOI: 10.1002/sta4.405.

70. BODNAR, Cristian; CANGEA, Cătălina; LIÒ, Pietro. Deep Graph Mapper: Seeing Graphs Through the Neural Lens. *Frontiers in Big Data*. 2021, vol. 4, p. 38. ISSN 2624-909X. Available from DOI: 10.3389/fdata.2021.680535.

71. HAJIJ, Mustafa; ROSEN, Paul; WANG, Bei. *Mapper on Graphs*. 2019. Available also from: https://github.com/USFDataVisualization/MapperOnGraphs.

72. DŁOTKO, Paweł. *BallMapper: The Ball Mapper Algorithm*. 2019. Available also from: https://cran.r-project.org/web/packages/BallMapper/index.html.

73. DLOTKO, Pawel; QIU, Wanling; RUDKIN, Simon. *Financial Ratios and Stock Returns Reappraised Through a Topological Data Analysis Lens*. 2019. Available from arXiv: 1911.10297.

74. DLOTKO, Pawel; RUDKIN, Simon; QIU, Wanling. *An Economic Topology of the Brexit vote*. 2019. Available from arXiv: 1909.03490.

75. DLOTKO, Pawel; RUDKIN, Simon; QIU, Wanling. *Topologically Mapping the Macroeconomy*. 2019. Available from arXiv: 1911.10476.

76. DLOTKO, Pawel; RUDKIN, Simon. *Visualising the Evolution of English Covid-19 Cases with Topological Data Analysis Ball Mapper*. 2020. Available from arXiv: 2004.03282.

77. QIU, Wanling; RUDKIN, Simon; DŁOTKO, Paweł. Refining Understanding of Corporate Failure Through a Topological Data Analysis Mapping of Altman's Z-score Model. *Expert Systems with Applications*. 2020, vol. 156, p. 113475. ISSN 0957-4174. Available from DOI: 10.1016/j.eswa.2020.113475.

78. PAWEŁ, Dłotko; GURNARI, Davide; SAZDANOVIC, Radmila. *Knot Invariants and Their Relations: A Topological Perspective*. 2021. Available from arXiv: 2109.00831.

79. MOJARAD, Musa; NEJATIAN, Samad; PARVIN, Hamid; MOHAMMADPOOR, Majid. A Fuzzy Clustering Ensemble Based on Cluster Clustering and Iterative Fusion of Base Clusters. *Appl. Intell.* 2019, vol. 49, no. 7, pp. 2567–2581. Available from DOI: 10.1007/s10489-018-01397-x.

80. STREHL, Alexander; GHOSH, Joydeep. Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions. *J. Mach. Learn. Res.* 2003, vol. 3, no. null, pp. 583–617. ISSN 1532-4435. Available from DOI: 10.1162/153244303321897735.

81. YING, Rex; YOU, Jiaxuan; MORRIS, Christopher; REN, Xiang; HAMILTON, William L.; LESKOVEC, Jure. Hierarchical Graph Representation Learning with Differentiable Pooling. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Montréal, Canada: Curran Associates Inc., 2018, pp. 4805–4815. NIPS'18. Available also from: https://dl.acm.org/doi/10.5555/3327345.3327389.

82. BIANCHI, Filippo Maria; GRATTAROLA, Daniele; ALIPPI, Cesare. Mincut Pooling in Graph Neural Networks. In: *International Conference on Learning Representations (ICLR 2020)*. 2020. Available also from: https://openreview.net/forum?id=BkxfshNYwB.

83. BODNAR, Cristian; CANGEA, Cătălina; LIÒ, Pietro. *Deep Graph Mapper*. 2021. Available also from: https://github.com/crisbodnar/dgm.

84. JEITZINER, Rachel; CARRIÈRE, Mathieu; ROUGEMONT, Jacques; OUDOT, Steve; HESS, Kathryn; BRISKEN, Cathrin. Two-Tier Mapper, an Unbiased Topology-based Clustering Method for Enhanced Global Gene Expression Analysis. *Bioinformatics*. 2019, vol. 35, no. 18, pp. 3339–3347. ISSN 1367-4803. Available from DOI: 10.1093/bioinformatics/btz052.

85. HU, Jingliang; HONG, Danfeng; ZHU, Xiao Xiang. MIMA: MAPPER-induced Manifold Alignment for Semi-Supervised Fusion of Optical Image and Polarimetric SAR Data. *IEEE Transactions on Geoscience and Remote Sensing*. 2019, vol. 57, no. 11, pp. 9025–9040. Available from DOI: 10.1109/TGRS.2019.2924113.

86. CYRANKA, Jacek; GEORGES, Alexander; MEYER, David. Mapper Based Classifier. In: *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. 2019, pp. 1099–1106. Available from DOI: 10.1109/ICMLA.2019.00184.

87. JEITZINER, Rachel. *TTMap, Two-Tier Mapper: A Clustering Tool Based on Topological Data Analysis*. 2021. Available also from: http://www.bioconductor.org/packages/release/bioc/html/TTMap.html.

88. REAVEN, G. M.; MILLER, R. G. An Attempt to Define the Nature of Chemical Diabetes Using a Multidimensional Analysis. *Diabetologia*. 1979, vol. 16, no. 1, pp. 17–24. ISSN 0012186X. Available from DOI: 10.1007/BF00423145.

89. NICOLAU, Monica; LEVINE, Arnold J.; CARLSSON, Gunnar. 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*. 2011, vol. 108, no. 17, pp. 7265–7270. ISSN 0027-8424. Available from DOI: 10.1073/pnas.1102826108.

90. NIELSON, Jessica L.; COOPER, Shelly R.; YUE, John K.; SORANI, Marco D.; INOUE, Tomoo; YUH, Esther L.; MUKHERJEE, Pratik; PETROSSIAN, Tanya C.; PAQUETTE, Jesse; LUM, Pek Y.; CARLSSON, Gunnar E.; VASSAR, Mary J.; LINGSMA, Hester F.; GORDON, Wayne A.; VALADKA, Alex B.; OKONKWO, David O.; MANLEY, Geoffrey T.; FERGUSON, Adam R.; INVESTIGATORS, TRACK-TBI. Uncovering Precision Phenotype-biomarker Associations in Traumatic Brain Injury Using Topological Data Analysis. *PLOS ONE*. 2017, vol. 12, no. 3, pp. 1–19. Available from DOI: 10.1371/journal.pone.0169490.

91. BRUNO, Jennifer L.; ROMANO, David; MAZAIKA, Paul; LIGHTBODY, Amy A.; HAZLETT, Heather Cody; PIVEN, Joseph; REISS, Allan L. Longitudinal Identification of Clinically Distinct Neurophenotypes in Young Children With Fragile X Syndrome. *Proceedings of the National Academy of Sciences*. 2017, vol. 114, no. 40, pp. 10767–10772. ISSN 0027-8424. Available from DOI: 10.1073/pnas.1620994114.

92. DIVER, Sarah; RICHARDSON, Matt; HALDAR, Koirobi; GHEBRE, Michael A.; RAMSHEH, Mohammadali Y.; BAFADHEL, Mona; DESAI, Dhananjay; COHEN, Emma Suzanne; NEWBOLD, Paul; RAPLEY, Laura; RUGMAN, Paul; PAVORD, Ian D.; MAY, Richard D.; BARER, Michael; BRIGHTLING, Christopher.E. Sputum Microbiomic Clustering in Asthma and Chronic Obstructive Pulmonary Disease Reveals a Haemophilus-predominant Subgroup. *Allergy*. 2020, vol. 75, no. 4, pp. 808–817. Available from DOI: 10.1111/all.14058.

93. ALMGREN, Khaled; KIM, Minkyu; LEE, Jeongkyu. Extracting Knowledge From the Geometric Shape of Social Network Data Using Topological Data Analysis. *Entropy*. 2017, vol. 19, no. 7. Available from DOI: 10.3390/e19070360.

94. ALMGREN, Khaled; KIM, Minkyu; LEE, Jeongkyu. Mining Social Media Data Using Topological Data Analysis. In: *2017 IEEE International Conference on Information Reuse and Integration (IRI)*. 2017, pp. 144–153. Available from DOI: 10.1109/IRI.2017.41.

95. FEGED-RIVADENEIRA, Alejandro; ÁNGEL, Andrés; GONZÁLEZ-CASABIANCA, Felipe; RIVERA, Camilo. Malaria Intensity In Colombia By Regions And Populations. *PLOS ONE*. 2018, vol. 13, no. 9, pp. 1–28. Available from DOI: 10.1371/journal.pone.0203673.

96. HARKEMA, Sebastian S.; SCHULTZ, Christopher J.; BERNDT, Emily B.; BITZER, Phillip M. Geostationary Lightning Mapper Flash Characteristics of Electrified Snowfall Events.

Weather and Forecasting. 2019, vol. 34, no. 5, pp. 1571–1585. Available from DOI: 10.1175/WAF-D-19-0082.1.

97. HU, Jingliang; HONG, Danfeng; WANG, Yuanyuan; ZHU, Xiao Xiang. A Topological Data Analysis Guided Fusion Algorithm: Mapper-Regularized Manifold Alignment. In: *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*. 2019, pp. 2822–2825. Available from DOI: 10.1109/IGARSS.2019.8898471.

98. GUO, Wei; BANERJEE, Ashis G. Toward Automated Prediction of Manufacturing Productivity Based on Feature Selection Using Topological Data Analysis. In: *2016 IEEE International Symposium on Assembly and Manufacturing (ISAM)*. 2016, pp. 31–36. Available from DOI: 10.1109/ISAM.2016.7750716.

99. SARIKONDA, Ghanashyam; PETTUS, Jeremy; PHATAK, Sonal; SACHITHANANTHAM, Sowbarnika; MILLER, Jacqueline F.; WESLEY, Johnna D.; CADAG, Eithon; CHAE, Ji; GANESAN, Lakshmi; MALLIOS, Ronna; EDELMAN, Steve; PETERS, Bjoern; VON HERRATH, Matthias. CD8 T-cell Reactivity to Islet Antigens Is Unique to Type 1 While CD4 T-cell Reactivity Exists in Both Type 1 and Type 2 Diabetes. *Journal of Autoimmunity*. 2014, vol. 50, pp. 77–82. ISSN 0896-8411. Available from DOI: 10.1016/j.jaut.2013.12.003.

100. GUO, Wei; BANERJEE, Ashis G. Identification of Key Features Using Topological Data Analysis for Accurate Prediction of Manufacturing System Outputs. *Journal of Manufacturing Systems*. 2017, vol. 43, pp. 225–234. ISSN 0278-6125. Available from DOI: 10.1016/j.jmsy.2017.02.015.

101. FALSETTI, Lorenzo; RUCCO, Matteo; PROIETTI, Marco; VITICCHI, Giovanna; ZACCONE, Vincenzo; SCARPONI, Mattia; GIOVENALI, Laura; MORONCINI, Gianluca; NITTI, Cinzia; SALVI, Aldo. Risk Prediction of Clinical Adverse Outcomes With Machine Learning in a Cohort of Critically Ill Patients With Atrial Fibrillation. *Scientific Reports*. 2021, vol. 11, no. 1, p. 18925. ISSN 2045-2322. Available from DOI: 10.1038/s41598-021-97218-2.

102. COUDRIAU, Marc; LAHMADI, Abdelkader; FRANÇOIS, Jérôme. Topological Analysis and Visualisation of Network Monitoring Data: Darknet Case Study. In: *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*. 2016, pp. 1–6. Available from DOI: 10.1109/WIFS.2016.7823920.

103. SIDDIQUI, Salman; SHIKOTRA, Aarti; RICHARDSON, Matthew; DORAN, Emma; CHOY, David; BELL, Alex; AUSTIN, Cary D.; EASTHAM-ANDERSON, Jeffrey; HARGADON, Beverley; ARRON, Joseph R.; WARDLAW, Andrew; BRIGHTLING, Christopher E.; HEANEY, Liam G.; BRADDING, Peter. Airway Pathological Heterogeneity in Asthma: Visualization of Disease Microclusters Using Topological Data Analysis. *Journal of Allergy and Clin-*

ical Immunology. 2018, vol. 142, no. 5, pp. 1457–1468. ISSN 0091-6749. Available from DOI: 10.1016/j.jaci.2017.12.982.

104. WANG, Tongxin; JOHNSON, Travis; ZHANG, Jie; HUANG, Kun. Topological Methods for Visualization and Analysis of High Dimensional Single-cell RNA Sequencing Data. In: *Biocomputing 2019*. 2018, pp. 350–361. Available from DOI: 10.1142/9789813279827_0032.

105. PHINYOMARK, Angkoon; IBÁÑEZ-MARCELO, Esther; PETRI, Giovanni. Resting-state fMRI Functional Connectivity: Big Data Preprocessing Pipelines and Topological Data Analysis. *IEEE Transactions on Big Data*. 2017, vol. 3, no. 4, pp. 415–428. Available from DOI: 10.1109/TBDATA.2017.2734883.

106. DUMAN, Ali Nabi; TATAR, Ahmet Emin; PIRIM, Harun. Uncovering Dynamic Brain Reconfiguration in MEG Working Memory n-Back Task Using Topological Data Analysis. *Brain Sciences*. 2019, vol. 9, no. 6. ISSN 2076-3425. Available from DOI: 10.3390/brainsci9060144.

107. CHITWOOD, Daniel H.; EITHUN, Mitchell; MUNCH, Elizabeth; OPHELDERS, Tim. Topological Mapper for 3D Volumetric Images. In: BURGETH, Bernhard; KLEEFELD, Andreas; NAEGEL, Benoît; PASSAT, Nicolas; PERRET, Benjamin (eds.). *Mathematical Morphology and Its Applications to Signal and Image Processing*. Cham: Springer International Publishing, 2019, pp. 84–95. ISBN 978-3-030-20867-7.

108. ROSEN, Paul; HAJIJ, Mustafa; TU, Junyi; ARAFIN, Tanvirul; PIEGL, Les. Inferring Quality in Point Cloud-based 3D Printed Objects Using Topological Data Analysis. *Computer-Aided Design and Applications*. 2019, vol. 16, pp. 519–527. Available from DOI: 10.14733/cadaps.2019.519-527.

109. WANG, Ziqi. Exploration of Topological Data Analysis In 3D Printing. In: *2020 International Conference on Information Science, Parallel and Distributed Systems (ISPDS)*. 2020, pp. 150–153. Available from DOI: 10.1109/ISPDS51347.2020.00038.

110. PEARSON, Paul; MÜLLNER, Daniel; SINGH, Gurjeet. *TDAmapper: Analyze High-dimensional Data Using Discrete Morse Theory*. 2015. Available also from: https://cran.r-project.org/web/packages/TDAmapper/index.html.

111. MÜLLNER, Daniel; BABU, Aravindakshan. *Python Mapper: An Open-source Toolchain for Data Exploration, Analysis, and Visualization*. 2013. Available also from: http://danifold.net/mapper.

112. SAUL, Nathaniel; VEEN, Hendrik Jacob van. *MLWave/kepler-mapper: 186f*. Zenodo, 2017. Version 1.0.1. Available from DOI: 10.5281/zenodo.1054444.

113. VEEN, Hendrik Jacob van; SAUL, Nathaniel; EARGLE, David; MANGHAM, Sam W. Kepler Mapper: a Flexible Python Implementation of the Mapper Algorithm. *Journal of Open Source Software*. 2019, vol. 4, no. 42, p. 1315. Available from DOI: 10.21105/joss.01315.

114. VEEN, Hendrik Jacob van; SAUL, Nathaniel; EARGLE, David; MANGHAM, Sam W. *Kepler Mapper*. 2019. Available also from: https://kepler-mapper.scikit-tda.org.

115. PIEKENBROCK, Matt; DORAN, Derek; KRAMER, Ryan. *Efficient Multi-scale Simplicial Complex Generation for Mapper*. 2018. Tech. rep. Available also from: https://peekxc.github.io/resources/indexed_mapper.pdf.

116. PIEKENBROCK, Matt. *Mapper*. 2019. Available also from: https://peekxc.github.io/Mapper.

117. GENIESSE, Caleb; SPORNS, Olaf; PETRI, Giovanni; SAGGAR, Manish. *DyNeuSR: Dynamical Neuroimaging Spatiotemporal Representations*. 2019. Available also from: https://braindynamicslab.github.io/dyneusr.

118. TAUZIN, Guillaume; LUPO, Umberto; TUNSTALL, Lewis; PÉREZ, Julian Burella; CAORSI, Matteo; MEDINA-MARDONES, Anibal M.; DASSATTI, Alberto; HESS, Kathryn. giotto-tda: A Topological Data Analysis Toolkit for Machine Learning and Data Exploration. *Journal of Machine Learning Research*. 2021, vol. 22, no. 39, pp. 1–6. Available also from: http://jmlr.org/papers/v22/20-325.html.

119. TAUZIN, Guillaume; LUPO, Umberto; TUNSTALL, Lewis; PÉREZ, Julian Burella; CAORSI, Matteo; MEDINA-MARDONES, Anibal M.; DASSATTI, Alberto; HESS, Kathryn. *giotto-tda*. 2021. Available also from: https://giotto.ai.

120. ZHOU, Youjia; CHALAPATHI, Nithin; RATHORE, Archit; ZHAO, Yaodong; WANG, Bei. Mapper Interactive: A Scalable, Extendable, and Interactive Toolbox for the Visual Exploration of High-Dimensional Data. In: *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)*. 2021, pp. 101–110. Available from DOI: 10.1109/PacificVis52677.2021.00021.

121. ZHOU, Youjia; CHALAPATHI, Nithin; RATHORE, Archit; ZHAO, Yaodong; WANG, Bei. *Mapper Interactive*. 2021. Available also from: https://mapperinteractive.github.io.

122. WALSH, Kieran; VOINEAGU, Mircea A; VAFAEE, Fatemeh; VOINEAGU, Irina. TDAview: An Online Visualization Tool for Topological Data Analysis. *Bioinformatics*. 2020, vol. 36, no. 18, pp. 4805–4809. ISSN 1367-4803. Available from DOI: 10.1093/bioinformatics/btaa600.

123. WALSH, Kieran; TAOUK, Kamile. *TDAview*. 2020. Available also from: https://github.com/Voineagulab/TDAview.

124. PEDREGOSA, Fabian; VAROQUAUX, Gaël; GRAMFORT, Alexandre; MICHEL, Vincent; THIRION, Bertrand; GRISEL, Olivier; BLONDEL, Mathieu; PRETTENHOFER, Peter; WEISS, Ron; DUBOURG, Vincent; VANDERPLAS, Jake; PASSOS, Alexandre; COURNAPEAU, David; BRUCHER, Matthieu; PERROT, Matthieu; DUCHESNAY, Édouard. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011, vol. 12, pp. 2825–2830. Available also from: `http://jmlr.org/papers/v12/pedregosa11a.html`.

125. BUITINCK, Lars; LOUPPE, Gilles; BLONDEL, Mathieu; PEDREGOSA, Fabian; MUELLER, Andreas; GRISEL, Olivier; NICULAE, Vlad; PRETTENHOFER, Peter; GRAMFORT, Alexandre; GROBLER, Jaques; LAYTON, Robert; VANDERPLAS, Jake; JOLY, Arnaud; HOLT, Brian; VAROQUAUX, Gaël. API Design for Machine Learning Software: Experiences From the Scikit-learn Project. In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 2013, pp. 108–122. Available from arXiv: `1309.0238`.

126. SAUL, Nathaniel; TRALIE, Chris. *Scikit-TDA: Topological Data Analysis for Python*. 2019. Available from DOI: `10.5281/zenodo.2533369`.

127. SAUL, Nathaniel; TRALIE, Chris. *Scikit-TDA*. 2019. Available also from: `https://scikit-tda.org`.

128. ZOMORODIAN, Afra. Fast Construction of the Vietoris-Rips Complex. *Computers & Graphics*. 2010, vol. 34, no. 3, pp. 263–271. ISSN 0097-8493. Available from DOI: `10.1016/j.cag.2010.03.007`.

129. ARAFAT, Naheed Anjum; BASU, Debabrota; BRESSAN, Stéphane. Topological Data Analysis with ϵ-net Induced Lazy Witness Complex. In: HARTMANN, Sven; KÜNG, Josef; CHAKRAVARTHY, Sharma; ANDERST-KOTSIS, Gabriele; TJOA, A Min; KHALIL, Ismail (eds.). *Database and Expert Systems Applications - 30th International Conference, DEXA 2019, Linz, Austria, August 26-29, 2019, Proceedings, Part II*. Springer, 2019, vol. 11707, pp. 376–392. Lecture Notes in Computer Science. Available from DOI: `10.1007/978-3-030-27618-8_28`.

130. CHAZAL, Frédéric; SILVA, Vin de; OUDOT, Steve. Persistence Stability for Geometric Complexes. *Geometriae Dedicata*. 2014, vol. 173, no. 1, pp. 193–214. ISSN 1572-9168. Available from DOI: `10.1007/s10711-013-9937-z`.

131. BELCHÍ, Francisco; BRODZKI, Jacek; BURFITT, Matthew; NIRANJAN, Mahesan. A Numerical Measure of the Instability of Mapper-type Algorithms. *Journal of Machine Learning Research*. 2020, vol. 21, no. 202, pp. 1–45. Available also from: `http://jmlr.org/papers/v21/19-540.html`.

132. CARRIÈRE, Mathieu; OUDOT, Steve. Structure and Stability of the One-dimensional Mapper. *Foundations of Computational Mathematics*. 2018, vol. 18, no. 6, pp. 1333–1396. ISSN 16153383. Available from DOI: `10.1007/s10208-017-9370-z`.

133. CARRIÈRE, Mathieu; MICHEL, Bertrand; OUDOT, Steve. Statistical Analysis and Parameter Selection for Mapper. *Journal of Machine Learning Research*. 2018, vol. 19, no. 12, pp. 1–39. Available also from: http://jmlr.org/papers/v19/17-291.html.

134. CORNELL, Filip. Using Topological Autoencoders as a Filtering Function for Global and Local Topology. In: *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*. 2020. Available also from: https://openreview.net/forum?id=0V6WLosuIfJ.

135. MCCABE, Michael. *Mapper Comparison with Wasserstein Metrics*. 2018. Available from arXiv: 1812.06232.

136. LOUGHREY, Ciara Frances; JUREK-LOUGHREY, Anna; ORR, Nick; DLOTKO, Pawel. Hotspot Identification for Mapper Graphs. In: *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*. 2020. Available also from: https://openreview.net/forum?id=reLv5jl2adC.

137. CAWI, Eric; LA ROSA, Patricio S.; NEHORAI, Arye. Designing Machine Learning Workflows With an Application to Topological Data Analysis. *PLOS ONE*. 2019, vol. 14, no. 12, pp. 1–26. Available from DOI: 10.1371/journal.pone.0225577.

138. RIIHIMÄKI, Henri; CHACHÓLSKI, Wojciech; THEORELL, Jakob; HILLERT, Jan; RAMANUJAM, Ryan. A Topological Data Analysis Based Classification Method for Multiple Measurements. *BMC Bioinformatics*. 2020, vol. 21, no. 1, p. 336. ISSN 1471-2105. Available from DOI: 10.1186/s12859-020-03659-3.

139. HAJIJ, Mustafa; ISTVAN, Kyle. *A Topological Framework for Deep Learning*. 2021. Available from arXiv: 2008.13697.

140. KUMARI, Nupur; R., Siddarth; RUPELA, Akash; GUPTA, Piyush; KRISHNAMURTHY, Balaji. ShapeVis: High-Dimensional Data Visualization at Scale. In: *Proceedings of The Web Conference 2020*. Taipei, Taiwan: Association for Computing Machinery, 2020, pp. 2920–2926. WWW '20. ISBN 9781450370233. Available from DOI: 10.1145/3366423.3380058.

141. VEEN, Hendrik Jacob van. Novel Topological Shapes of Model Interpretability. In: *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*. 2020. Available also from: https://openreview.net/forum?id=G-kWQ9WvBMq.

142. RAMAMURTHI, Yashwanth; AGARWAL, Tripti; CHATTOPADHYAY, Amit. A Topological Similarity Measure between Multi-resolution Reeb Spaces. *IEEE Transactions on Visualization and Computer Graphics*. 2021. Available from DOI: 10.1109/TVCG.2021.3087273. (Early Access).

143. RAMOS, Christian; CALUS, Mario; SCHOKKER, Dirkjan. Persistence of Functional Microbiota Composition Across Generations. *Scientific Reports*. 2021, vol. 11, no. 1, p. 19007. ISSN 2045-2322. Available from DOI: 10.1038/s41598-021-98097-3.

144. SAJJADI, Seyed Erfan; DRAGHI, Barbara; SACCHI, Lucia; DAGLIANI, Arianna; HOLMES, John; TUCKER, Allan. Building Trajectories Over Topology with TDA-PTS: An Application in Modelling Temporal Phenotypes of Disease. In: KOPRINSKA, Irena; KAMP, Michael; APPICE, Annalisa; LOGLISCI, Corrado; ANTONIE, Luiza; ZIMMERMANN, Albrecht; GUIDOTTI, Riccardo; ÖZGÖBEK, Özlem; RIBEIRO, Rita P.; GAVALDÀ, Ricard; GAMA, João; ADILOVA, Linara; KRISHNAMURTHY, Yamuna; FERREIRA, Pedro M.; MALERBA, Donato; MEDEIROS, Ibéria; CECI, Michelangelo; MANCO, Giuseppe; MASCIARI, Elio; RAS, Zbigniew W.; CHRISTEN, Peter; NTOUTSI, Eirini; SCHUBERT, Erich; ZIMEK, Arthur; MONREALE, Anna; BIECEK, Przemyslaw; RINZIVILLO, Salvatore; KILLE, Benjamin; LOMMATZSCH, Andreas; GULLA, Jon Atle (eds.). *ECML PKDD 2020 Workshops*. Cham: Springer International Publishing, 2020, pp. 48–61. ISBN 978-3-030-65965-3. Available from DOI: 10.1007/978-3-030-65965-3_4.

145. ALJANOBI, Fatima Ali; LEE, Jeongkyu. Topological Data Analysis for Classification of Heart Disease Data. In: *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*. 2021, pp. 210–213. Available from DOI: 10.1109/BigComp51126.2021.00047.

146. TANG, Yunbo; CHEN, Dan; LI, Xiaoli. Dimensionality Reduction Methods for Brain Imaging Data Analysis. *ACM Comput. Surv.* 2021, vol. 54, no. 4. ISSN 0360-0300. Available from DOI: 10.1145/3448302.

147. SANTOS, Fernando A. N.; RAPOSO, Ernesto P.; COUTINHO-FILHO, Maurício D.; COPELLI, Mauro; STAM, Cornelis J.; DOUW, Linda. Topological Phase Transitions in Functional Brain Networks. *Phys. Rev. E*. 2019, vol. 100, p. 032414. Available from DOI: 10.1103/PhysRevE.100.032414.

148. MATHEWS, James C; POURYAHYA, Maryam; MOOSMÜLLER, Caroline; KEVREKIDIS, Yannis G; DEASY, Joseph O; TANNENBAUM, Allen. Molecular Phenotyping Using Networks, Diffusion, and Topology: Soft Tissue Sarcoma. *Scientific Reports*. 2019, vol. 9, no. 1, p. 13982. ISSN 2045-2322. Available from DOI: 10.1038/s41598-019-50300-2.

149. LUO, Ping; CHEN, Bolin; LIAO, Bo; WU, Fang-Xiang. Predicting Disease-associated Genes: Computational Methods, Databases, and Evaluations. *WIREs Data Mining and Knowledge Discovery*. 2021, vol. 11, no. 2, e1383. Available from DOI: 10.1002/widm.1383.

150. GUO, Muhua. Application of Topological Data Analysis in Co-infections and Its Effectiveness. In: *2020 International Conference on Big Data and Social Sciences (ICBDSS)*. 2020, pp. 15–18. Available from DOI: 10.1109/ICBDSS51270.2020.00011.

151. BALLESTEROS, Nathalia; MUÑOZ, Marina; PATIÑO, Luz Helena; HERNÁNDEZ, Carolina; GONZÁLEZ-CASABIANCA, Felipe; CARROLL, Iván; SANTOS-VEGA, Mauricio; CASCANTE, Jaime; ANGEL, andrés; FEGED-RIVADENEIRA, Alejandro; PALMA-

CUERO, Mónica; FLÓREZ, Carolina; GOMEZ, Sergio; GUCHTE, Adriana van de; KHAN, Zenab; DUTTA, Jayeeta; OBLA, Ajay; ALSHAMMARY, Hala Alejel; GONZALEZ-REICHE, Ana S.; HERNANDEZ, Matthew M.; SORDILLO, Emilia Mia; SIMON, Viviana; BAKEL, Harm van; PANIZ-MONDOLFI, Alberto E.; RAMÍREZ, Juan David. Deciphering the Introduction and Transmission of SARS-CoV-2 in the Colombian Amazon Basin. *PLOS Neglected Tropical Diseases*. 2021, vol. 15, no. 4, pp. 1–22. Available from DOI: 10.1371/journal.pntd.0009327.

152. KAVEH-YAZDY, Fatemeh; ZARIFZADEH, Sajjad. Track Iran's National Covid-19 Response Committee's Major Concerns Using Two-stage Unsupervised Topic Modeling. *International Journal of Medical Informatics*. 2021, vol. 145, p. 104309. ISSN 1386-5056. Available from DOI: 10.1016/j.ijmedinf.2020.104309.

153. TOPAZ, Chad M.; ZIEGELMEIER, Lori; HALVERSON, Tom. Topological Data Analysis of Biological Aggregation Models. *PLOS ONE*. 2015, vol. 10, no. 5, pp. 1–26. Available from DOI: 10.1371/journal.pone.0126383.

154. MURUGAN, Jeff; ROBERTSON, Duncan. *An Introduction to Topological Data Analysis for Physicists: From LGM to FRBs*. 2019. Available from arXiv: 1904.11044.

155. ZHANG, Kexin; WU, Jiasheng; YOO, Hyeonsuk; LEE, Yongjin. Machine Learning-based Approach for Tailor-made Design of Ionic Liquids: Application to CO_2 Capture. *Separation and Purification Technology*. 2021, vol. 275, p. 119117. ISSN 1383-5866. Available from DOI: 10.1016/j.seppur.2021.119117.

156. SMITH, Alexander D.; DŁOTKO, Paweł; ZAVALA, Victor M. Topological Data Analysis: Concepts, Computation, and Applications in Chemical Engineering. *Computers & Chemical Engineering*. 2021, vol. 146, p. 107202. ISSN 0098-1354. Available from DOI: 10.1016/j.compchemeng.2020.107202.

157. KIM, Hannah; VOGEL, Christian. Deciphering Active Wildfires in the Southwestern USA Using Topological Data Analysis. *Climate*. 2019, vol. 7, no. 12. ISSN 2225-1154. Available from DOI: 10.3390/cli7120135.

158. OHANUBA, F.O.; ISMAIL, M.T.; ALI, M.K. Majahar. Topological Data Analysis via Unsupervised Machine Learning for Recognizing Atmospheric River Patterns on Flood Detection. *Scientific African*. 2021, vol. 13, e00968. ISSN 2468-2276. Available from DOI: 10.1016/j.sciaf.2021.e00968.

159. RIVERA-CASTRO, Rodrigo; NAZAROV, Ivan; XIANG, Yuke; MAKSIMOV, Ivan; PLETNEV, Aleksandr; BURNAEV, Evgeny. An Industry Case of Large-scale Demand Forecasting of Hierarchical Components. In: *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. 2019, pp. 134–139. Available from DOI: 10.1109/ICMLA.2019.00029.

160. MAJUMDAR, Sourav; LAHA, Arnab Kumar. Clustering and Classification of Time Series Using Topological Data Analysis With Applications to Finance. *Expert Systems with Applications*. 2020, vol. 162, p. 113868. ISSN 0957-4174. Available from DOI: 10.1016/j.eswa.2020.113868.

161. KNUDSON, Angélica; GONZÁLEZ-CASABIANCA, Felipe; FEGED-RIVADENEIRA, Alejandro; PEDREROS, Maria Fernanda; APONTE, Samanda; OLAYA, Adriana; CASTILLO, Carlos F; MANCILLA, Elvira; PIAMBA-DORADO, Anderson; SANCHEZ-PEDRAZA, Ricardo; SALAZAR-TERREROS, Myriam Janeth; LUCCHI, Naomi; UDHAYAKUMAR, Venkatachalam; JACOB, Chris; PANCE, Alena; CARRASQUILLA, Manuela; APRÁEZ, Giovanni; ANGEL, Jairo Andrés; RAYNER, Julian C; CORREDOR, Vladimir. Spatio-Temporal Dynamics of Plasmodium Falciparum Transmission Within a Spatial Unit on the Colombian Pacific Coast. *Scientific Reports*. 2020, vol. 10, no. 1, p. 3756. ISSN 2045-2322. Available from DOI: 10.1038/s41598-020-60676-1.

162. BENEDETTI-CECCHI, Lisandro. Complex Networks of Marine Heatwaves Reveal Abrupt Transitions in the Global Ocean. *Scientific Reports*. 2021, vol. 11, no. 1, p. 1739. ISSN 2045-2322. Available from DOI: 10.1038/s41598-021-81369-3.

163. HENSEL, Felix; MOOR, Michael; RIECK, Bastian. A Survey of Topological Machine Learning Methods. *Frontiers in Artificial Intelligence*. 2021, vol. 4, p. 52. ISSN 2624-8212. Available from DOI: 10.3389/frai.2021.681108.

164. KITANISHI, Yoshitake; FUJIWARA, Masakazu; BINKOWITZ, Bruce. Patient Journey Through Cases of Depression From Claims Database Using Machine Learning Algorithms. *PLOS ONE*. 2021, vol. 16, no. 2, pp. 1–11. Available from DOI: 10.1371/journal.pone.0247059.

165. NAITZAT, Gregory; ZHITNIKOV, Andrey; LIM, Lek-Heng. Topology of Deep Neural Networks. *Journal of Machine Learning Research*. 2020, vol. 21, no. 184, pp. 1–40. Available also from: http://jmlr.org/papers/v21/20-345.html.

166. AMER, Mohammed; MAUL, Tomás. A Review of Modularization Techniques in Artificial Neural Networks. *Artificial Intelligence Review*. 2019, vol. 52, no. 1, pp. 527–561. ISSN 1573-7462. Available from DOI: 10.1007/s10462-019-09706-7.

167. KHAN, Asifullah; SOHAIL, Anabia; ZAHOORA, Umme; QURESHI, Aqsa Saeed. A Survey of the Recent Architectures of Deep Convolutional Neural Networks. *Artificial Intelligence Review*. 2020, vol. 53, no. 8, pp. 5455–5516. ISSN 1573-7462. Available from DOI: 10.1007/s10462-020-09825-6.

168. RATHORE, Archit; CHALAPATHI, Nithin; PALANDE, Sourabh; WANG, Bei. TopoAct: Visually Exploring the Shape of Activations in Deep Learning. *Computer Graphics Forum*. 2021, vol. 40, no. 1, pp. 382–397. Available from DOI: 10.1111/cgf.14195.

169. GOLDFARB, Daniel. *Understanding Deep Neural Networks Using Topological Data Analysis*. 2018. Available from arXiv: 1811.00852.

170. ELHAMDADI, Hamza; CANAVAN, Shaun; ROSEN, Paul. AffectiveTDA: Using Topological Data Analysis to Improve Analysis and Explainability in Affective Computing. *IEEE Transactions on Visualization and Computer Graphics*. 2021. Available from DOI: 10.1109/TVCG.2021.3114784. (Early Access).

171. GABELLA, Maxime. Topology of Learning in Feedforward Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*. 2021, vol. 32, no. 8, pp. 3588–3592. Available from DOI: 10.1109/TNNLS.2020.3015790.

172. RAVISHANKER, Nalini; CHEN, Renjie. *Topological Data Analysis (TDA) for Time Series*. 2019. Available from arXiv: 1909.10604.

173. UMEDA, Y.; KANEKO, J.; KIKUCHI, H. Topological Data Analysis and Its Application to Time-series Data Analysis. *Fujitsu Scientific and Technical Journal*. 2019, vol. 55, pp. 65–71. Available also from: https://www.fujitsu.com/global/documents/about/resources/publications/fstj/archives/vol55-2/paper15.pdf.

174. DAGLIATI, Arianna; GEIFMAN, Nophar; PEEK, Niels; HOLMES, John H.; SACCHI, Lucia; BELLAZZI, Riccardo; SAJJADI, Seyed Erfan; TUCKER, Allan. Using Topological Data Analysis and Pseudo Time Series to Infer Temporal Phenotypes From Electronic Health Records. *Artificial Intelligence in Medicine*. 2020, vol. 108, p. 101930. ISSN 0933-3657. Available from DOI: 10.1016/j.artmed.2020.101930.

175. SILVA, Vanessa Freitas; SILVA, Maria Eduarda; RIBEIRO, Pedro; SILVA, Fernando. Time Series Analysis via Network Science: Concepts and Algorithms. *WIREs Data Mining and Knowledge Discovery*. 2021, vol. 11, no. 3, e1404. Available from DOI: 10.1002/widm.1404.

176. TUKPAH, Ann-Marcia C; CAWI, Eric; WOLF, Laurie; NEHORAI, Arye; CUMMINGS-VAUGHN, Lenise. Development of an Institution-specific Readmission Risk Prediction Model for Real-time Prediction and Patient-centered Interventions. *Journal of General Internal Medicine*. 2021. ISSN 1525-1497. Available from DOI: 10.1007/s11606-020-06549-9.

177. WU, Chengyuan; HARGREAVES, Carol Anne. Topological Machine Learning for Multivariate Time Series. *Journal of Experimental & Theoretical Artificial Intelligence*. 2021, vol. 0, no. 0, pp. 1–16. Available from DOI: 10.1080/0952813X.2021.1871971.

178. HUANG, Wei-Ming; HONG, Tzung-Pei; LAN, Guo-Cheng; CHIANG, Ming-Chao; LIN, Jerry Chun-Wei. Temporal-based Fuzzy Utility Mining. *IEEE Access*. 2017, vol. 5, pp. 26639–26652. Available from DOI: 10.1109/ACCESS.2017.2774510.

179. HUANG, Wei-Ming; HONG, Tzung-Pei; CHIANG, Ming-Chao; LIN, Jerry Chun-Wei. Using Multi-conditional Minimum Thresholds in Temporal Fuzzy Utility Mining. *International Journal of Computational Intelligence Systems*. 2019, vol. 12, pp. 613–626. ISSN 1875-6883. Available from DOI: 10.2991/ijcis.d.190426.001.

180. ROUSSEEUW, Peter J. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics*. 1987, vol. 20, pp. 53–65. ISSN 0377-0427. Available from DOI: 10.1016/0377-0427(87)90125-7.

181. ANDREWS, David F.; HERZBERG, A.M. *Data: A Collection of Problems from Many Fields for the Student and Research Worker*. Springer New York, 1985. Springer Series in Statistics. ISBN 9783540961253. Available from DOI: 10.1007/978-1-4612-5098-2.

182. VEER, Laura J van't; DAI, Hongyue; VIJVER, Marc J van de; HE, Yudong D; HART, Augustinus A M; MAO, Mao; PETERSE, Hans L; KOOY, Karin van der; MARTON, Matthew J; WITTEVEEN, Anke T; SCHREIBER, George J; KERKHOVEN, Ron M; ROBERTS, Chris; LINSLEY, Peter S; BERNARDS, René; FRIEND, Stephen H. Gene Expression Profiling Predicts Clinical Outcome of Breast Cancer. *Nature*. 2002, vol. 415, no. 6871, pp. 530–536. ISSN 1476-4687. Available from DOI: 10.1038/415530a.

183. YAGER, Ronald R.; REFORMAT, Marek Z.; TO, Nhuan D. Drawing on the Ipad to Input Fuzzy Sets With an Application to Linguistic Data Science. *Information Sciences*. 2019, vol. 479, pp. 277–291. ISSN 0020-0255. Available from DOI: 10.1016/j.ins.2018.11.048.

184. BYSTROV, Dmitriy; OLIMJON, Toirov; GULZODA, Mustafakulova; DILFUZA, Yakubova. Fuzzy Systems for Computational Linguistics and Natural Language. In: *Proceedings of the 3rd International Conference on Networking, Information Systems & Security*. Marrakech, Morocco: Association for Computing Machinery, 2020. NISS2020. ISBN 9781450376341. Available from DOI: 10.1145/3386723.3387873.

185. YE, Jin; ZHAN, Jianming; XU, Zeshui. A Novel Decision-making Approach Based on Three-way Decisions in Fuzzy Information Systems. *Information Sciences*. 2020, vol. 541, pp. 362–390. ISSN 0020-0255. Available from DOI: 10.1016/j.ins.2020.06.050.

186. HERRERA-VIEDMA, Enrique; PALOMARES, Iván; LI, Cong-Cong; CABRERIZO, Francisco Javier; DONG, Yucheng; CHICLANA, Francisco; HERRERA, Francisco. Revisiting Fuzzy and Linguistic Decision Making: Scenarios and Challenges for Making Wiser Decisions in a Better Way. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2021, vol. 51, no. 1, pp. 191–208. Available from DOI: 10.1109/TSMC.2020.3043016.

187. WU, Tsu-Yang; LIN, Jerry Chun-Wei; YUN, Unil; CHEN, Chun-Hao; SRIVASTAVA, Gautam; LV, Xianbiao. An Efficient Algorithm for Fuzzy Frequent Itemset Mining. *Journal of Intelligent & Fuzzy Systems*. 2020, vol. 38, pp. 5787–5797. ISSN 1875-8967. Available from DOI: 10.3233/JIFS-179666.

188. LI, Li-xuan; HUO, Ying; LIN, Jerry Chun-Wei. Cross-dimension Mining Model of Public Opinion Data in Online Education Based on Fuzzy Association Rules. *Mobile Networks and Applications*. 2021. ISSN 1572-8153. Available from DOI: 10.1007/s11036-021-01769-7.

189. LIN, Jerry Chun-Wei; LI, Ting; FOURNIER-VIGER, Philippe; HONG, Tzung-Pei; SU, Ja-Hwung. Fast algorithms for mining multiple fuzzy frequent itemsets. In: *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. 2016, pp. 2113–2119. Available from DOI: 10.1109/FUZZ-IEEE.2016.7737952.

190. NAGARAJ, S; MOHANRAJ, E. A Novel Fuzzy Association Rule for Efficient Data Mining of Ubiquitous Real-time Data. *Journal of Ambient Intelligence and Humanized Computing*. 2020, vol. 11, no. 11, pp. 4753–4763. ISSN 1868-5145. Available from DOI: 10.1007/s12652-020-01736-2.

191. WU, Jimmy Ming-Tai; SRIVASTAVA, Gautam; WEI, Min; YUN, Unil; LIN, Jerry Chun-Wei. Fuzzy High-utility Pattern Mining in Parallel and Distributed Hadoop Framework. *Information Sciences*. 2021, vol. 553, pp. 31–48. ISSN 0020-0255. Available from DOI: 10.1016/j.ins.2020.12.004.

192. ALFARO-GARCÍA, Víctor G; MERIGÓ, José M; PEDRYCZ, Witold; GÓMEZ MONGE, Rodrigo. Citation Analysis of Fuzzy Set Theory Journals: Bibliometric Insights About Authors and Research Areas. *International Journal of Fuzzy Systems*. 2020, vol. 22, no. 8, pp. 2414–2448. ISSN 2199-3211. Available from DOI: 10.1007/s40815-020-00924-8.

193. YANG, Cheng-Hong; CHUANG, Li-Yeh; LIN, Yu-Da. An Improved Fuzzy Set-based Multifactor Dimensionality Reduction for Detecting Epistasis. *Artificial Intelligence in Medicine*. 2020, vol. 102, p. 101768. ISSN 0933-3657. Available from DOI: 10.1016/j.artmed.2019.101768.

194. GOUR, Alekh; PARDASANI, K.R. Type II Fuzzy Set-based Data Analytics to Explore Amino Acid Associations in Protein Sequences of Swine Influenza Virus. 2020, vol. 88, p. 105856. ISSN 1568-4946. Available from DOI: 10.1016/j.asoc.2019.105856.

195. CROSS, Valerie; ZMUDA, Michael; PAUL, Rahul; HALL, Lawrence. Fuzzy Set Similarity for Feature Selection in Classification. In: *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. 2020, pp. 1–8. Available from DOI: 10.1109/FUZZ48607.2020.9177820.

196. CAI, Yuliang; ZHANG, Huaguang; SUN, Shaoxin; WANG, Xianchang; HE, Qiang. Axiomatic Fuzzy Set Theory-based Fuzzy Oblique Decision Tree With Dynamic Mining Fuzzy Rules. *Neural Computing and Applications*. 2020, vol. 32, no. 15, pp. 11621–11636. ISSN 1433-3058. Available from DOI: 10.1007/s00521-019-04649-0.

197. LIN, Song-Shun; SHEN, Shui-Long; ZHOU, Annan; XU, Ye-Shuang. Risk Assessment and Management of Excavation System Based on Fuzzy Set Theory and Machine Learning Methods. *Automation in Construction*. 2021, vol. 122, p. 103490. ISSN 0926-5805. Available from DOI: 10.1016/j.autcon.2020.103490.

198. ZHANG, Mengsen; SAGGAR, Manish. Complexity of intrinsic brain dynamics shaped by multiscale structural constraints. *bioRxiv*. 2020. Available from DOI: 10.1101/2020.05.14.097196.

199. VENDRAMIN, L.; NALDI, M. C.; CAMPELLO, R. J. G. B. Fuzzy Clustering Algorithms and Validity Indices for Distributed Data. In: *Partitional Clustering Algorithms*. Ed. by CELEBI, M. Emre. Cham: Springer International Publishing, 2015, pp. 147–192. ISBN 978-3-319-09259-1. Available from DOI: 10.1007/978-3-319-09259-1_5.

200. BEN-DAVID, Shai; LUXBURG, Ulrike von; PÁL, Dávid. A Sober Look at Clustering Stability. In: LUGOSI, Gabor; SIMON, Hans Ulrich (eds.). *Learning Theory*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 5–19. ISBN 978-3-540-35296-9. Available from DOI: 10.1007/11776420_4.

201. BEN-HUR, Asa; ELISSEEFF, Andre; GUYON, Isabelle. A Stability Based Method for Discovering Structure in Clustered Data. In: *Biocomputing 2002*. 2001, pp. 6–17. Available from DOI: 10.1142/9789812799623_0002.

202. LUXBURG, Ulrike von. Clustering Stability: An Overview. *Found. Trends Mach. Learn.* 2010, vol. 2, no. 3, pp. 235–274. ISSN 1935-8237. Available from DOI: 10.1561/2200000008.

203. STONE, M. Cross-validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*. 1974, vol. 36, no. 2, pp. 111–133. Available from DOI: 10.1111/j.2517-6161.1974.tb00994.x.

204. CAMPELLO, R.J.G.B.; HRUSCHKA, E.R. A Fuzzy Extension of the Silhouette Width Criterion for Cluster Analysis. *Fuzzy Sets and Systems*. 2006, vol. 157, no. 21, pp. 2858–2875. ISSN 0165-0114. Available from DOI: 10.1016/j.fss.2006.07.006.

205. GUO, Chenjuan; MA, Yu; YANG, Bin; JENSEN, Christian S.; KAUL, Manohar. EcoMark: Evaluating Models of Vehicular Environmental Impact. In: *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. Redondo Beach, California: Association for Computing Machinery, 2012, pp. 269–278. SIGSPATIAL '12. ISBN 9781450316910. Available from DOI: 10.1145/2424321.2424356.

206. BLACKARD, Jock A.; DEAN, Denis J. Comparative Accuracies of Artificial Neural Networks and Discriminant Analysis in Predicting Forest Cover Types From Cartographic Variables. *Computers and Electronics in Agriculture*. 1999, vol. 24, no. 3, pp. 131–151. ISSN 0168-1699. Available from DOI: 10.1016/S0168-1699(99)00046-0.

207. AKBULUT, Yaman; ŞENGÜR, Abdulkadir; GUO, Yanhui; POLAT, Kemal. KNCM: Kernel Neutrosophic *C*-Means Clustering. *Applied Soft Computing*. 2017, vol. 52, pp. 714–724. ISSN 1568-4946. Available from DOI: 10.1016/j.asoc.2016.10.001.

208. LOUGHREY, Ciara Frances; JUREK-LOUGHREY, Anna; ORR, Nick; DLOTKO, Pawel. Hotspot identification for Mapper graphs. In: *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*. 2020. Available also from: https://openreview.net/forum?id=reLv5jl2adC.

209. KANNAN, Harish; SAUCAN, Emil; ROY, Indrava; SAMAL, Areejit. Persistent Homology of Unweighted Complex Networks via Discrete Morse Theory. *Scientific Reports*. 2019, vol. 9, no. 1, p. 13817. ISSN 2045-2322. Available from DOI: 10.1038/s41598-019-50202-3.

210. KARTUN-GILES, Alexander P.; BIANCONI, Ginestra. Beyond the Clustering Coefficient: A Topological Analysis of Node Neighbourhoods in Complex Networks. *Chaos, Solitons & Fractals: X*. 2019, vol. 1, p. 100004. ISSN 2590-0544. Available from DOI: 10.1016/j.csfx.2019.100004.

211. HERNÁNDEZ SERRANO, Daniel; HERNÁNDEZ-SERRANO, Juan; SÁNCHEZ GÓMEZ, Darío. Simplicial Degree in Complex Networks. Applications of Topological Data Analysis to Network Science. *Chaos, Solitons & Fractals*. 2020, vol. 137, p. 109839. ISSN 0960-0779. Available from DOI: 10.1016/j.chaos.2020.109839.

A Publications by Author

[P1] **Q.-T. Bui**, B. Vo, H.-A. N. Do, N. Q. V. Hung, and V. Snasel. F-Mapper: A Fuzzy Mapper Clustering Algorithm. *Knowledge-Based Systems*, vol. 189, 105107, 2020.

- Available from DOI: 10.1016/j.knosys.2019.105107
- Cited by (Google Scholar): 11
- Journal information (WoS 2020): SCIE, Q1, IF = 8.038, Rank = 16/139 (2nd decile)

[P2] S. G. Quek, G. Selvachandran, F. Smarandache, J. Vimala, S. H. Le, **Q.-T. Bui**, and V. C. Gerogiannis. Entropy Measures for Plithogenic Sets and Applications in Multi-attribute Decision Making. *Mathematics*, vol. 8, no. 6, 965, 2020.

- Available from DOI: 10.3390/math8060965
- Cited by (Google Scholar): 7
- Journal information (WoS 2020): SCIE, Q1, IF = 2.258, Rank = 24/330 (1st decile)

[P3] **Q.-T. Bui**, B. Vo, V. Snasel, W. Pedrycz, T.-P. Hong, N.-T. Nguyen, and M.-Y. Chen. SFCM: a Fuzzy Clustering Algorithm of Extracting the Shape Information of Data. *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 1, pp. 75–89, 2021.

- Available from DOI: 10.1109/TFUZZ.2020.3014662
- Cited by (Google Scholar): 9
- Journal information (WoS 2020): SCIE, Q1, IF = 12.029, Rank = 4/139 (1st decile)

[P4] J. S. Chai, G. Selvachandran, F. Smarandache, V. C. Gerogiannis, L. H. Son, **Q.-T. Bui**, and B. Vo. New Similarity Measures for Single-valued Neutrosophic Sets With Applications in Pattern Recognition and Medical Diagnosis Problems. *Complex & Intelligent Systems*, vol. 7, no. 2, pp. 703–723, 2021.

- Available from DOI: 10.1007/s40747-020-00220-w
- Cited by (Google Scholar): 10
- Journal information (WoS 2020): SCIE, Q2, IF = 4.927, Rank = 36/139

B Publications by Author Which Are Currently Under Review

[P5] **Q.-T. Bui**, M.-P. Ngo, V. Snasel, W. Pedrycz, and B. Vo. The Sequence of Neutrosophic Soft Sets and a Decision-Making Problem in Medical Diagnosis. *International Journal of Fuzzy Systems*.

- Status: Under Review (Round 2)
- Submission time: July 2021
- Journal information (WoS 2020): SCIE, Q1, IF = 4.673, Rank = 35/161

[P6] **Q.-T. Bui**, M.-P. Ngo, V. Snasel, W. Pedrycz, and B. Vo. Information Measures on Neutrosophic Fuzzy Sets in Making the Multi-criteria Decision. *IEEE Transactions on Fuzzy Systems*

- Status: Under Review
- Submission time: September 2021
- Journal information (WoS 2020): SCIE, Q1, IF = 12.029, Rank = 4/139 (1st decile)

[P7] **Q.-T. Bui**, V. Snasel, W. Pedrycz, and B. Vo. The Mapper Algorithm: An Outstanding Representative of Topological Data Analysis. *ACM Computing Surveys*.

- Status: Under Review
- Submission time: November 2021
- Journal information (WoS 2020): SCIE, Q1, IF = 10.282, Rank = 4/110 (1st decile)

[P8] T.T.D. Nguyen, L.T.T. Nguyen, **Q.-T. Bui**, U. Yun, and B. Vo. An Efficient Topological-Based Clustering Method on Spatial Data in Network Space. *Knowledge-Based Systems*.

- Status: Under Review
- Submission time: December 2021
- Journal information (WoS 2020): SCIE, Q1, IF = 8.038, Rank = 16/139 (2nd decile)

[P9] T.B.T. Tran, M.-P. Ngo, **Q.-T. Bui**, V. Snasel, and B. Vo. Another Approach for Operations on Neutrosophic Soft Sets Based on the Novel Norms for Constructing Topological Structures. *AIMS Mathematics*.

- Status: Under Review
- Submission time: December 2021
- Journal information (WoS 2020): SCIE, Q2, IF = 1.427, Rank = 83/330

C Summary of the Author's Academic Activities During the Doctoral Course

1. Subjects

   - BIO-INSPIRED COMPUTING (460-6017/02)
     Completion time: Academic year 2017/2018
   - APPLIED LINEAR ALGEBRA (401-1328/01)
     Completion time: Academic year 2017/2018
   - DATA ANALYSIS (460-6016/02)
     Completion time: Academic year 2018/2019
   - SOCIAL NETWORKING (460-6018/02)
     Completion time: Academic year 2018/2019
   - PARALLEL ALGORITHMS (460-6008/02)
     Completion time: Academic year 2019/2020
   - ENGLISH LANGUAGE DR. (712-0191/02)
     Completion time: Academic year 2019/2020

2. Publications

   - Number of published papers (Web of Science, 2020): 4, in which two are Q1 (1st decile), one is Q1 (2st decile), and one is Q2
   - Number of under review manuscripts (Web of Science): 5

3. Courses

   - DATA SCIENTIST WITH PYTHON
     Organization: DataCamp
     Completion time: January 2018
     Online certificate: datacamp.com/statement-of-accomplishment/track/30c6dcb1afd31325a7771d9fe2e26c6f3c76544c
   - PYTHON CREATIVE PROGRAMMING
     Organization: VNUHCM - University of Science
     Completion time: September 2019
     Online certificate: github.com/quangthinhbui/certificates
   - IBM DATA SCIENCE
     Organization: International Business Machines Corporation (IBM)
     Completion time: March 2020
     Online certificate: coursera.org/verify/professional-cert/CAL8HXT93UFN

- STATISTICS WITH PYTHON
  Organization: University of Michigan
  Completion time: March 2020
  Online certificate: coursera.org/verify/specialization/Z2LPFTSS64W9
- MATHEMATICS FOR MACHINE LEARNING
  Organization: Imperial College London
  Completion time: April 2020
  Online certificate: coursera.org/verify/specialization/L7VQQK23BYYQ
- MATHEMATICS FOR DATA SCIENCE
  Organization: National Research University Higher School of Economics
  Completion time: April 2020
  Online certificate: coursera.org/verify/specialization/UGBVNNJPH8MJ
- MATHS FOR DATA SCIENCE
  Organization: Vietnam Institute for Advanced Study in Mathematics
  Completion time: December 2020
  Online certificate: github.com/quangthinhbui/certificates
- BLOCKCHAIN MATHEMATICS AND COMPUTING
  Organization: Vietnam Institute for Advanced Study in Mathematics
  Completion time: July 2021
  Online certificate: github.com/quangthinhbui/certificates

4. Workshops

- SCIENCE COMMUNICATION & STEM
  Organization(s): National University of Singapore & Vietnam National University, Hanoi
  Workshop time: November 2019
  Online certificate: github.com/quangthinhbui/certificates
- STRENGTHENING GLOBAL CITIZENSHIP IN HIGHER EDUCATION FOR SUSTAINABLE DEVELOPMENT
  Organization(s): SEAMEO Regional Training Center
  Workshop time: October 2020
  Online certificate: github.com/quangthinhbui/certificates