SFCM: A Fuzzy Clustering Algorithm of Extracting the Shape Information of Data

Quang-Thinh Bui, Bay Vo, Vaclav Snasel, *Member, IEEE*, Witold Pedrycz, *Fellow, IEEE*, Tzung-Pei Hong, *Senior Member, IEEE*, Ngoc-Thanh Nguyen, *Senior Member, IEEE*, and Mu-Yen Chen

***Abstract*—Topological data analysis is a new theoretical trend using topological techniques to mine data. This approach helps determine topological data structures. It focuses on investigating the global shape of data rather than on local information of high-dimensional data. The Mapper algorithm is considered as a sound representative approach in this area. It is used to cluster and identify concise and meaningful global topological data structures that are out of reach for many other clustering methods. In this article, we propose a new method called the Shape Fuzzy *C*-Means (SFCM) algorithm, which is constructed based on the Fuzzy *C*-Means algorithm with particular features of the Mapper algorithm. The SFCM algorithm can not only exhibit the same clustering ability as the Fuzzy *C*-Means but also reveal some relationships through visualizing the global shape of data supplied by the Mapper. We present a formal proof and include experiments to confirm our claims. The performance of the enhanced algorithm is demonstrated through a comparative analysis involving the original algorithm, Mapper, and the other fuzzy set based improved algorithm, F-Mapper, for synthetic and real-world data. The comparison is conducted with respect to output visualization in the topological sense and clustering stability.**

***Index Terms*—Big data, fuzzy clustering, Fuzzy *C*-Means (FCM), Mapper, shape of data, topological data analysis (TDA).**

Manuscript received March 2, 2020; revised May 21, 2020 and July 15, 2020; accepted July 22, 2020. Date of publication August 6, 2020; date of current version December 30, 2020. *(Corresponding author: Bay Vo.)*

Quang-Thinh Bui is with the Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam, and also with the Faculty of Electrical Engineering and Computer Science, VŠB-Technical University of Ostrava, 708 00 Ostrava-Poruba, Czech Republic (e-mail: qthinhbui@gmail.com).

Bay Vo is with the Faculty of Information Technology, Ho Chi Minh City University of Technology (HUTECH), Ho Chi Minh City 700000, Vietnam (e-mail: vd.bay@hutech.edu.vn).

Vaclav Snasel is with the Faculty of Electrical Engineering and Computer Science, VŠB-Technical University of Ostrava, 708 00 Ostrava-Poruba, Czech Republic (e-mail: vaclav.snasel@vsb.cz).

Witold Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2R3, Canada, with the Department of Electrical and Computer Engineering, Faculty of Engineering, King Abgudulaziz University, Jeddah 21589, Saudi Arabia, and also with the Systems Research Institute, Polish Academy of Sciences, Warsaw 01-224, Poland (e-mail: wpedrycz@ualberta.ca).

Tzung-Pei Hong is with the Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan, and also with the Department of Computer Science and Engineering, National Sun Yat-Sen University, Kaohsiung 804, Taiwan (e-mail: tphong@nuk.edu.tw).

Ngoc-Thanh Nguyen is with the Department of Applied Informatics, Wroclaw University of Science and Technology, Wrocław 50-370, Poland (e-mail: ngoc-thanh.nguyen@pwr.edu.pl).

Mu-Yen Chen is with the Department of Engineering Science, National Cheng Kung University, Tainan City 701, Taiwan (e-mail: mychen119@gs.ncku.edu.tw).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TFUZZ.2020.3014662

I. Introduction

BIG DATA has emerged as one of the most noteworthy research pursuits in recent years, according to the statistics produced for this term returned by Google search [1]. The area is concerned with handling a massive collection of data whose extraction, analysis, visualization, sharing, storing, transfer, and search come with a high level of complexity. Besides, it grows at an ever-increasing rate. Volume, velocity, and variety (3V) are the three key characteristics of big data identified in the original research report published by the META Group [2]. Since then, other key characteristics have also been supplemented, including veracity, value [3], validity, volatility [4], and variability [5].

Nowadays, there is a vast amount of data being created and collected from several resources every day. The traditional data processing tools cannot afford to manage huge and complex data. Therefore, it is critical to develop effective and efficient techniques for processing big data. Modern data science uses geometry and topology as the natural tools for analyzing massive amounts of data. Geometry is used mainly as quantitative mathematics, while topology, on the other hand, provides a formal language for a qualitative approach [6]. Implementing topology in data analysis, known as topological data analysis (TDA), has been developed as a new and promising area of data mining research during the past two decades. TDA is considered a pretreatment process to find the structural features of datasets before proceeding with further supervised or unsupervised analysis. In TDA, a summary or a compressed representation of all data features is created to help rapidly uncover data characteristics [7], [8]. Its purpose is not to replace existing techniques but to provide a new and convenient approach in data mining.

TDA aims to provide additional tools for analyzing data in engineering and science. It applies concepts coming from geometric topology to reveal some relationships present in data [6]. Topology, focusing on shape and shape invariants through continuous deformations, has been marked an important development in data analysis in the transition from theory to applications [7]–[9]. The successful strategy of TDA comes from a very intuitive idea, "Data has shape, shape has meaning, meaning drives value" [10]. The data shape creates the first useful impression that helps the analyst choose suitable methods for data. This shape is usually considered one of the first highlights, and topology emphasizes it as being the most important aspect of data [11]. Some proposed TDA approaches focus on the data shape and can be implemented effectively to handle high-dimensional data [6].

Besides the very powerful techniques built on persistent homology [12]–[14], the Mapper algorithm is one of the most effective TDA tools to identify topological shape characteristics of data. It aims to visualize the topological and geometric details of the high-dimensional datasets in the form of a graph. It is a formal tool to extract simple but significant descriptors with the qualitative analysis, simplification, and visualization of these complex data [15]. This algorithm was first developed by Singh *et al*. [15] based on the Morse theory [16]. The Mapper algorithm creates a topological approximation for a metric space by mapping it via a continuous function to another space of a lower dimensionality. In the simplest way, the continuous function is 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. In the general form, this algorithm can return an abstract simplicial complex that is considered an easy and convenient method to visualize a topological summary of data.

The Mapper algorithm has two classic applications, such as clustering [15], [17]–[19] and feature selection [15], [18], [20]–[22]. In clustering, the advantages have been manifested more effectively than those by some classical clustering techniques, such as single-linkage, k-means, and principal component analysis (PCA) [15], [18], [20]. Moreover, the clustering performance of the Mapper algorithm highlights the effectiveness of this approach in clearly identifying various meaningful subgroups.

From a theoretical perspective, the Mapper algorithm is still a fuzzy clustering algorithm, with a visualization ability to extract the shape summary of data. The results of the algorithm are still very sensitive to selection 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 lens 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, we propose a new algorithm, called the Shape Fuzzy *C*-Means (SFCM) algorithm, based on the two algorithms, the Fuzzy C-Means (FCM) and the Mapper. The SFCM algorithm is constructed based on a bright and significant representative of overlapping clustering by carefully combining it with the advantages of TDA. This algorithm brings two simultaneous possibilities for data: Fuzzy clustering as with the FCM and shape detection as with the Mapper. It is proposed as a summarization technique to transform large, complex data into informative representations and to provide interactive visualizations for their exploration. In particular, from the Mapper perspective, this algorithm significantly reduces the dependence on the parameters. Like the original methods, this approach is expected to solve in a useful and robust manner some problems encountered in many fields, especially in bioinformatics and neuroscience.

In the experimental part of this study, four synthetic and real-world datasets, namely the Unit Circle, the Two Concentric Circles with Noise, the 3-D Trefoil Knot, and the Reaven and Miller Diabetes, have been experimented with by running the SFCM algorithm. The results are evaluated and contrasted with those produced by the two algorithms, namely Mapper and F-Mapper, in three ways, including the visualization from the topological standpoint, the clustering stability through matching coefficient, and the internal index with silhouette coefficient, to demonstrate the efficiency of the new algorithm. This method can generate similar outputs from the topological view as the previous two methods. At the same time, the clustering stability and internal index of the SFCM algorithm are better than those of the Mapper and F-Mapper algorithms in most experimental cases. Furthermore, the SFCM is also piloted for some large datasets of high dimensionality to demonstrate the working prospect with big data of this algorithm.

In general, we summarize contributions of this article as follows.

1) We propose a new algorithm, named SFCM, with two simultaneous possibilities for mining data: Fuzzy clustering as with the FCM and shape detection as with the Mapper.
2) The SFCM can generate similar outputs from the topological standpoint as the previous two methods, Mapper and F-Mapper.
3) The clustering stability and internal index of the SFCM are better than those produced by the Mapper and F-Mapper in most experimental cases.
4) The SFCM can visualize highly complex data in a simple, meaningful, and informative form with the potential of applicability to big data.

The rest of this article is organized as follows. Section II elaborates on the brief background of two algorithms, FCM and Mapper. Section III introduces the SFCM algorithm. In Section IV, experiments with the SFCM algorithm involving four publicly available datasets are described, and the results are evaluated with those delivered by the Mapper and F-Mapper algorithms in three aspects including the output visualization, the clustering stability, and the internal index. Finally, Section V concludes this article.

II. Background

This section offers a brief introduction to the underlying notation and ideas of the FCM algorithm and the Mapper algorithm leading to the emergence of the SFCM algorithm. More specific details could be found in the textbooks [23], [24].

A. Fuzzy C-Means Clustering

Nowadays, fuzzy sets [25] have become an increasingly powerful tool to model situations where vagueness and uncertainty of data exist, such as decision-making [26], [27], frequent itemset mining [28], [29], linguistics [30], [31], web mining [32], [33], bioinformatics [34], [35], and so on.

While the k-means, also referred to as hard C-means, is a popular algorithm in data clustering, the FCM algorithm is an important vehicle to cope with overlapping clustering. The clustering idea of the FCM algorithm was first proposed in 1973 by Dunn [36], and its convergence was later improved in 1981 by Bezdek [37].

Given a finite dataset $X = \{x_1, x_2, \ldots, x_n\}$ with n elements, the FCM algorithm organizes these data into c clusters characterized by some centroids $C = \{v_0, v_1, \ldots, v_c\}$. The main task of this algorithm is to iteratively optimize (minimize)

Algorithm 1: Fuzzy C-Means Algorithm.

Input	- Finite dataset X with n elements, - Number of clusters c, - Fuzzification exponent m, - Termination criteria: k_{max} or $\varepsilon \in (0;1)$.
Ouput	- Fuzzy partition matrix $U = [u_{ij}]$, - Cluster centroid matrix $C = [v_j]$.
Method	1: $k = 0$. 2: Initializing U. 3: **repeat** 3.1: Calculating the cluster centroid matrix by using the formula (3). 3.2: Updating the fuzzy partition matrix $U^{(k)}$ by using the formula (4). 3.3: $k = k + 1$. 4: **until** $\max_{i,j}\{\lvert u_{ij}^{(k+1)} - u_{ij}^{(k)}\rvert\} < \varepsilon$ or $k = k_{max}$.

the following objective function:

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

where the membership degree u_{ij} of the point x_i in the cluster j is constrained in the following way: $u_{ij} \in [0;1]$ and

$$\sum_{j=1}^{c} u_{ij} = 1, 1 \le i \le n. \tag{2}$$

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

The fuzzification exponent m plays an important role in this algorithm. The best choice for this exponent is made experimentally [38]. For most data types, the value of *m* is indicated to be the best lies within $[1.5,\ 2.5]$, the best value is obtained through evaluation of some cluster validity indices [39]. The majority of researchers have assigned this parameter to a fixed value, $m = 2.0$, based on empirical studies. In 2008, Pedrycz and Oliveira [40] gave experimental evidence behind the selection of the fuzzification exponent at this value. The optimal values of the fuzzification coefficient are typically lower than the commonly used value of 2.0. However, in this study, we set the value of the fuzzy exponents 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 (membership degrees) u_{ij} by the following formulas:

$$v_j = \frac{\sum_{i=1}^{n}(u_{ij})^m x_i}{\sum_{i=1}^{n}(u_{ij})^m} \tag{3}$$

and

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|}\right)^{\frac{2}{m-1}}}. \tag{4}$$

The iterative algorithm terminates once the following termination condition has been satisfied:

$$\max_{i,j}\left\{\left|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right|\right\} < \varepsilon \in (0;1) \tag{5}$$

or

$$k = k_{\max} \tag{6}$$

where $k_{\max}$ is a given positive integer (number of iterations).

B. Mapper Algorithm

The Mapper algorithm was first introduced in 2007 by Singh *et al*. [15] as a computational method to extract simple descriptions of high-dimensional datasets in the form of simplicial complexes. It has increasingly become one of the fundamental tools of TDA to study the topological structure and shape of data. In view of topology, the shape of the data is coordinate invariant, deformation invariant, and represented in a compressed form with meaningful and useful qualitative features [18]. This algorithm prudently exploits the characteristic properties of topology that are often used in data analysis. In general, this method is similar to other TDA algorithms when converting data properties from discrete to continuous. However, it summarizes, visualizes, and explores a given dataset X in a metric space by establishing the nerve of a refined pull-back cover of X through a well-chosen continuous function f, called *lens*. This nerve is created from the refining process on the cover of X that is the inverse image of a cover U on the filter range $f(X)$ through the lens f. The cover U divides the filter range into N equal intervals with the same overlapping percentage *p*, which are called the *resolution parameters*. The resolution parameters exhibit a certain correlation with the number of nodes and the connectivity between nodes in a nerve [41]. The clustering algorithm aims to separate each constituent of the pull-back cover into connected components for creating the refined pull-back cover. The resolution parameters will significantly affect the output nerve [41], and the Mapper algorithm is not affected by any specific clustering algorithm [11].

This method then has enjoyed tremendous success as a useful and robust tool in data science, especially in bioinformatics [17], [18], [20]–[22], [42]–[44]. It has been applied successfully in the analysis of some simulation data for biomolecular folding pathways and opens a promising direction for structural information in the biomolecular folding problems [17]. Nicolau *et al*. [20] combined the Mapper algorithm with disease-specific genomic analysis to identify a subgroup of breast cancers with a unique mutational profile and excellent survival rates. Notably, Lum *et al*. [18] extracted insights from the shape of complex data using this algorithm for identifying the finer stratifications of breast cancer patients, the voting patterns of the House of Representatives, and the playing styles of the NBA players. Moreover, the Mapper algorithm is used to exploit meaningful discoveries in preclinical spinal cord injury and traumatic brain injury [21], with the same study concluding that data-driven discovery using TDA has a great potential application for decision support for basic research and clinical problems such as outcome assessment, neurocritical care, treatment planning, and rapid precise diagnosis [21]. Besides the outstanding developments in bioinformatics, this approach is also used in social networks [19], [45] and neuroscience [42], [43], [46]. In recent years, the Mapper algorithm has specially enlarged in the field of neuroscience, where the output of this algorithm is used to

Algorithm 2: Mapper Algorithm.

Input	- A dataset X in a metric space, - A lens $f: X \to \mathbb{R}$, - The resolution parameters: Numbers N of equal intervals and same p overlapping percentage, - A clustering algorithm.
Output	A simplicial complex as a geometric visualization of the dataset.
Method	1: Mapping the dataset X to the real number line $\mathbb{R}$ using the lens f; 2: Enveloping the filter range $f(X)$ by a cover $\mathcal{U}$ including N equal intervals and the same p overlapping percentage; 3: For each $U \in \mathcal{U}$, separating $f^{-1}(U)$ into clusters $C_{U,1}, C_{U,2}, \ldots, C_{U,i_U}$ by applying a clustering algorithm or partitioning connected components; 4: Establishing the nerve of the refined pull-back cover of X determined by the clusters: - The node $v_{U,i}$ for each cluster $C_{U,i}$, - The edge between $v_{U,i}$ and $v_{U',j}$ if and only if $C_{U,i} \cap C_{U',j} \neq \emptyset$, - Node color: Average of lens values of points in the node (red is high and blue is low), - Node size: Number of points in the node.

visualize the brain [42], [43]. Overall, it has become a useful and robust TDA tool to solve problems in many fields.

In addition to these applications, the Mapper algorithm has been improved in the choice of the parameters, and, thus, several versions suitable for different usages were produced. The theoretical foundations needed to evaluate this algorithm have also been developed. Dey *et al*. [47], [48] offered the Multiscale Mapper algorithm that uses a tower of covers instead of a single one in the original method. This version has been examined with regard to its structure and stability by computing the persistence diagrams and has been shown to be an efficient and practical algorithm. Jeitziner *et al*. [49] developed the Two-Tier Mapper to cluster global gene expression data in order to detect subgroups and identify their distinguishing features. There is no requirement from any user that could induce bias in this version because all the parameters are data-driven. Dłotko [50] proposed a new Mapper-inspired algorithm called the Ball Mapper algorithm. It was described as encapsulating both the local and the global structures of a dataset rigorously and could thus be used in exploratory data analysis. This algorithm was used to examine the economic topology of the Brexit vote [51] and to visualize the point cloud of financial ratios as an abstract 2-D graph [52]. Bui *et al*. [41] proposed the F-Mapper algorithm based on the effectiveness of the fuzzy set based clustering techniques. The FCM algorithm was used to optimize the resolution parameters by automatically dividing the intervals with arbitrary overlapping percentages when covering the filter range.

In the recent years, the theoretical frameworks of the Mapper algorithm have been developed alongside practical applications [53], [54]. Carrière and Oudot [53] proposed a theoretical framework to guarantee the structure of the Mapper algorithm, its stability, and convergence to the Reeb graph. Through some further statistical analysis, the research of Carrière *et al*. [54] showed that the Mapper is an optimal estimator of the Reeb graph. This provides a method to automatically tune the parameters and obtain confidence regions for the topological features on the persistence diagram, which may be used to identify reliable Mapper outputs.

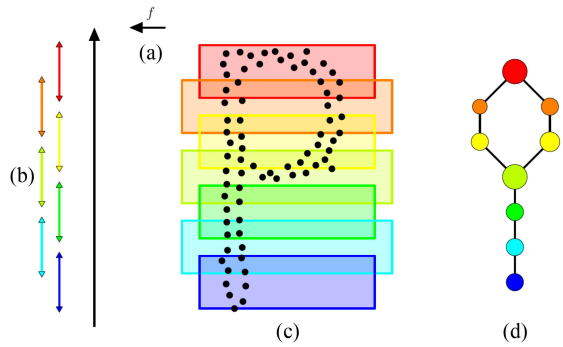


Fig. 1. Illustration of the Mapper algorithm on a point cloud with P shape. (a) Mapping all data points to the vertical line that illustrates the real number set. (b) Using the equal intervals with the same overlapping percentage to cover the filter range ($N = 7$ and $p = 30\%$). (c) Separating each element of the pull-back cover into clusters by using the single-linkage clustering. (d) Establishing the nerve of clusters regarded as a refined pull-back cover. The average of lens values of points in each node is displayed by colors: Red is high and blue is low. The size of each node instructs the number of points in this node.

Furthermore, the Mapper algorithm has also been developed with many different applied orientations in mind. Hajij *et al*. [55] studied parallel analysis in the construction of this algorithm and also used the Mapper construction to graph visualization [56]. This application provides a sound theoretical basis for summarizing network data while preserving the core structures. Cyranka *et al*. [57] proposed a classifier based on applying the Mapper algorithm to data projected onto a latent space, which was obtained by using PCA or autoencoders. This method has great robustness when compared to traditional convolutional neural networks with deep learning.

Unless otherwise specified, the Mapper algorithm that is chosen to examine throughout this article has the simplest form with a continuous real-valued function in the lens role. An example of how the Mapper algorithm can be used to gain insight into a point cloud data is illustrated in Fig. 1 with a specific and clear description.

III. SFCM Algorithm

The SFCM algorithm is introduced as a fuzzy clustering algorithm for extracting simple and meaningful description of high-dimensional datasets. This algorithm is constructed by carefully combining the advantages of the two algorithms, the FCM and Mapper. With regard to 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. Compared to the Mapper algorithm, the parameter choice with 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 working performance in certain ways.

Algorithm 3: Shape Fuzzy C-Means Algorithm.

Input	- A finite dataset X in a metric space, - Number of clusters c, - Fuzzification exponent m, be often set $m = 2.0$, - Termination criteria: k_{max} or $\varepsilon \in (0; 1)$, be often set = $k_{max} = 10000$ and $\varepsilon = 0.0001$ - Overlapping threshold τ.
Ouput	A simplicial complex as a geometric visualization of the dataset.
Method	1: Clustering the dataset X by using the FCM algorithm: 1.1: $k = 0$ 1.2: Initializing U. 1.3: **repeat** 1.3.1: Calculating the cluster centroid matrix by using (3). 1.3.2: Updating the fuzzy partition matrix $U^{(k)}$ by using (4). 1.3.3: $k = k + 1$. 1.4: **until** $\max_{i,j}\{\lvert u_{ij}^{(k+1)} - u_{ij}^{(k)}\rvert\} < \varepsilon$ or $k = k_{max}$. 1.5: Using the overlapping threshold τ to identify the c clusters from the fuzzy partition matrix U. 2: Establishing the nerve of the cover of X defined by the clusters: - The node v_i for each cluster C_i, - The edge between v_i and v_j if and only if $C_i \cap C_j \neq \emptyset$, - Node color: Average of colored values of points in the node (red is high and blue is low), - Node size: Number of points in the node.

The SFCM algorithm takes an input that is a dataset X located in a metric space. Due to the result of this combination, the parameters of this algorithm include the following.

1) Number of clusters c.
2) The fuzzification exponent m whose value has been often chosen as the fixed value $m = 2$ based on the previous empirical studies.
3) Termination criteria: $k_{\max}$ or $\varepsilon \in (0; 1)$.
4) The overlap threshold τ. It decides the overlap between clusters by comparing the membership degree of each data point.

The SFCM algorithm also creates a geometric simplicial complex as in the case of the Mapper algorithm. The number of clusters c and the overlapping threshold τ, called the resolution parameters, will much affect the output result. Each cluster is represented by a node, and each connection between two clusters is shown by an edge. The number of nodes is positively correlated with the number of clusters, and the connectivity in simplicial complex is negatively correlated 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. That is, 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) In the first process, the FCM algorithm organizes the points in the dataset X into the c clusters with the fuzzy partition matrix U determined by the membership degrees of the points for each cluster. The FCM algorithm stops when one of the termination criteria is satisfied. The clusters then serve as the covers for X, and the overlap 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 popular technique in fuzzy data mining [58], [59]. A data point belongs to a cluster if its membership degree for this cluster is greater than or equal to the value of the threshold τ. Therefore, it can belong to one or more clusters.
2) In the second process, the nerve of the cover of data X defined by the clusters is constructed as the Mapper algorithm does. Every pair of nodes corresponding to two clusters C_1 and C_2 is connected by an edge if and only if the intersection $C_1 \cap C_2$ of the two clusters is not empty.

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 $1 \leq i \leq n$. Set

$$T_0 = \min_{i,j} u_{ij} \tag{7}$$

and

$$T_1 = \min_{i} \max_{j} u_{ij}. \tag{8}$$

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 and 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 clusters. Therefore, the potential values of τ are limited to a close interval $[T_0; T_1]$, where 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.

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

$$\tau' \leq \tau \Rightarrow G_\tau \subseteq G_{\tau'}. \tag{9}$$

Proof: Because G_τ and $G_{\tau'}$ have the same vertices set V, $G_\tau \subseteq G_{\tau'} \Leftrightarrow \Sigma_\tau \subseteq \Sigma_{\tau'}$, we need to prove $\Sigma_\tau \subseteq \Sigma_{\tau'} \Leftrightarrow \tau \geq \tau'$.

For 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 threshold τ

$$x_k \in C_i^\tau \cap C_j^\tau \Rightarrow \begin{cases} x_k \in C_i^\tau \\ x_k \in C_j^\tau \end{cases} \Rightarrow \begin{cases} u_{ki} \geq \tau \\ u_{kj} \geq \tau \end{cases} \Rightarrow \begin{cases} u_{ki} \geq \tau' \\ u_{kj} \geq \tau' \end{cases}$$
$$\Rightarrow \begin{cases} x_k \in C_i^{\tau'} \\ x_k \in C_j^{\tau'} \end{cases} \Rightarrow x_k \in C_i^{\tau'} \cap C_j^{\tau'}.$$

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

Lemma 2: The nondecreasing 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 nondecreasing sequence of the threshold values. According to Lemma 1, there is a nondecreasing sequence of the output graph of the SFCM algorithm

$$\cdots \subseteq G_{\tau_{k+1}} \subseteq G_{\tau_k} \subseteq G_{\tau_{k-1}} \subseteq \cdots. \tag{10}$$

This sequence is a filtered simplicial complex. Therefore, using the algebraic topology knowledge, there is a sequence of

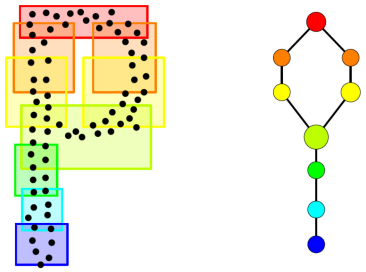


Fig. 2. Illustration of the SFCM algorithm on a point cloud with P shape. (a) FCM algorithm organizes all data points into nine clusters ($N = 9$) that are considered as the cover of the point cloud. (b) Establishing the nerve of the cover defined by the clusters as for the Mapper algorithm. The average of values of points through the projection on the vertical line in each node is displayed by colors: Red is high and blue is low. The size of each node instructs the number of points in this node.

homology groups that generates the filtered simplicial complex

$$\cdots \subseteq H\left(G_{\tau_{k+1}}\right) \subseteq H\left(G_{\tau_k}\right) \subseteq H\left(G_{\tau_{k-1}}\right) \subseteq \cdots . \quad (11)$$

Note that there are only finitely many threshold τ values where the structure of the output graph G_τ can change. As such, there is a finite nondecreasing sequence of the threshold values

$$0 \leq \ T_0 = \tau_0 \ \leq \tau_1 \leq \tau_2 \leq \cdots \leq \ \tau_l = T_1 \ \leq 1.$$

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

$$H\left(G_{\tau_l}\right) \subseteq H\left(G_{\tau_{l-1}}\right) \subseteq \cdots \subseteq H\left(G_{\tau_1}\right) \subseteq H\left(G_{\tau_0}\right). \quad (12)$$

Since the output of the algorithm is a graph, which is a simplicial complex with its dimension not greater than one, all the homology groups in formula (12) thus have a dimensionality not greater than one as well. The SFCM algorithm can be completely extended 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 those in the case of using the Čech complex or Vietoris–Rips complex to build the simplicial complex on the dataset. These create the premise for analyzing data using the persistent homology tool.

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

Lemma 3: The time complexity of the SFCM algorithm is $O(kcNn^2)$, 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 formulas (3), (4), and (1) require $O(cnN)$, $O(cnN^2)$, and $O(c^2n)$ operations, respectively [60]. So, the time complexity of the FCM algorithm is $O(k(cnN + cnN^2 + c^2n)) \rightarrow O(kcnN^2)$. In the second process, it takes at most $O(k(k-1))$ steps to create the connections between nodes. Overall, the time complexity of the SFCM algorithm is $O(kcnN^2 + k(k-1)) \rightarrow O(kcnN^2)$.

As an example, the SFCM algorithm is annotated in Fig. 2 with the same dataset used for the previous original algorithm in Fig. 1. In this case, the point cloud sampled with a P shape is first covered by the nine clusters that are obtained by the FCM algorithm. The graph is then composed of vertices, representing the clusters, and edges, representing the connections between the two nonintersecting clusters. The nodes are colored by the values of the data points through the projection on the vertical line. The high value is displayed in red, and the low value is displayed in blue.

TABLE I
COMPARISON BETWEEN THE ALGORITHMS: MAPPER, F-MAPPER, FCM, AND SFCM

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

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 high-dimensional data. The FCM algorithm takes an advantage of being efficient to create the cover for the whole space. This offers a new and sound idea of high feasibility because it creates the clusters in a natural way based on the density of data points. Moreover, the number of clusters helps the users proactively decide as to the complexity of the output representations. With this highlight, the SFCM algorithm can visualize highly complex data in a simple and meaningful form by a reasonable parameter selection depending on users' perspective. In addition, 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. This construction method connects naturally with visualization by providing a good theoretical foundation for simplifying large and complex data while preserving their core structures. Table I presents the comparison of the four algorithms, namely the Mapper, F-Mapper, FCM, and SFCM.

Through the examples illustrated above, the two algorithms, Mapper and SFCM, produce similar results. However, to confirm this assertion, some experiments are needed to proceed on actual datasets. Moreover, F-Mapper is also a fuzzy set based algorithm with a similar trend to the two mentioned algorithms. Therefore, the next section shows the experimental evaluations on the four real-world datasets for comparing the results among the three algorithms.

IV. EXPERIMENTAL STUDIES

Many software packages use the Mapper algorithm as a theoretical framework and are freely available in MATLAB, Python, R, and Spark [41]. In this article, the *KeplerMapper*, a free library implementing the Mapper algorithm in Python by Saul *et al*. and Veen *et al*. [61], [62], 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 has recently been developed and improved as a library in the Scikit-TDA project [63] that provides TDA Python tools in widely usable and easily approachable forms. We have extended and modified the code in the open-source codebase of the KeplerMapper to create our own version, the SFCM algorithm. Besides, we have also used the *SciKit-Fuzzy* package

to implement the experiments for the FCM algorithm. It is a library of fuzzy logic algorithms intended for use in the SciPy Stack by the Python computing language. If nothing changes, the termination criteria, reviewed in this article, are set as a constant $k_{\max} = 10\ 000$ and $\varepsilon = 0.0001 \in (0; 1)$.

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

For the output visualization, all the three algorithms, Mapper, F-Mapper, and SFCM, are processed sequentially on each real-world dataset.

1) The parameters of the Mapper algorithm, such as the lens f, number of intervals N, overlapping percentage p, and clustering algorithm, are set the same as in the previous famous works.
2) The parameters of the F-Mapper algorithm with regard to the lens f, number of intervals N, and clustering algorithm, have then been chosen the same as those with the Mapper algorithm. The overlapping threshold τ is assigned by a tested value for achieving a similar output to those of the Mapper algorithm from the topological viewpoint.
3) Finally, the parameters of the SFCM algorithm have been selected to show that it is possible to produce results that are quite similar to those of the Mapper algorithm from the topological viewpoint. The number of clusters N is set to be the same as the number of nodes in the output graph of the Mapper algorithm. The overlapping threshold τ is assigned with a specific value to get a similar output to the Mapper algorithm.

Note that in these experiments, the overlapping percentage p and the overlapping threshold τ have been chosen to have exactly two decimal places.

Clustering stability is considered as a main feature to confirm the validity of the sample-based algorithms [64]. In practice, it is widely used for optimizing the parameters of the algorithm. However, the theoretical analysis of this notion is quite limited, with only the Ben-Hur *et al*. [65], Luxburg [66], and Ben-David *et al.* [64] research works. In these experiments, we discuss clustering stability following the Ben-Hur *et al*. [65] view through the matching coefficient.

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

$$M_{ij} = \begin{cases} 0 \text{ if } x_i \text{ and } x_j \text{ in the same cluster and } i \neq j \\ 1 \text{ otherwise.} \end{cases} \tag{13}$$

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

$$L_1, L_2 = M^1, M^2 = \sum_{i,j} M^1_{ij} M^2_{ij}. \tag{14}$$

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

Given 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

$$MC\ (L_1, L_2) = \frac{n_{00} + n_{11}}{n_{00} + n_{01} + n_{10} + n_{11}}. \tag{15}$$

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

$$MC\ (L_1, L_2) = 1 - \frac{1}{n^2} M^1 - M^2, M^1 - M^2. \tag{16}$$

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 [66]. We conduct these experiments to compute the matching coefficient through the subsampling of the data detailed in [65]. The clustering stability is computationally estimated by the method of the k-fold cross validation as follows:

1) splitting the data into k subsamples by choosing $m, k \in \mathbb{N}$ such that $n = km$ and removing the m points from $m(i-1)+1$ to mi for each $1 \leq i \leq k$, then obtaining k subsamples with $(k-1)m$ points in each sample;
2) computing the matching coefficient between the clustering of each pair of subsamples on the $(k-2)m$ points of their intersection;
3) averaging the matching coefficients restricted to the subsamples by summing the coefficients and dividing it by $k(k-1)$.

The silhouette coefficient is a validation index of consistency within clusters of data [67]. A silhouette of a data point measures the clustering quality of the point. It compares the average distance of the point to all the elements in the same cluster with its average distance to all the elements in other clusters. The average computed for all the points of a whole dataset is a measure of the clustering quality.

In this article, we use a fuzzy version of the silhouette coefficient in the exclusive clustering [68] for evaluating these experiments. The silhouette $s(x_k)$ of a data point x_k is defined as

$$s\,(x_k) = \frac{b_{pk} - a_{pk}}{\max\{a_{pk}, b_{pk}\}} \tag{17}$$

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 nearest cluster prototype. Obviously, the silhouette $s(x_k)$ is in a closed interval $[-1,\ 1]$. The silhouette $s(X)$ of a dataset X is the average for all points of the dataset as follows:

$$s\,(X) = \frac{\sum_{k=1}^{n} (u_{pk} - u_{qk})\, s_k}{\sum_{k=1}^{n} (u_{pk} - u_{qk})} \tag{18}$$

where u_{pk} and u_{qk} are the first and the second largest elements of the kth column in the fuzzy partition matrix, respectively. Good partitions are expected to bring greater values to $s(x_k)$ and thus to $s(X)$ than bad ones. So, the silhouette coefficient under the fuzzy version is also a sound maximization index [60].

For clustering stability and internal index, all the three algorithms, Mapper, F-Mapper, and SFCM, are also processed sequentially on each real-world dataset.

1) The Mapper algorithm: First, the parameters are set as in the previous popular experiments for each dataset. After that, the overlapping percentage p is changed such that the output graphs have the same shape under the condition that the other parameters such as the lens, number of intervals, and clustering algorithm are fixed.
2) The F-Mapper algorithm: First, the parameters such as the lens, number of intervals, and clustering algorithm are identified invariably as those in the Mapper algorithm for each dataset. The overlapping threshold τ is changed to guarantee that the output graphs have the same shape as the Mapper algorithm from a topological perspective.
3) The SFCM algorithm: First, the number of clusters is chosen to be the same as the number of nodes of the Mapper algorithm output, and the threshold is selected to get a similar output to the Mapper from a topological perspective for 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 by using the method of k-fold cross validation for each case in which overlapping percentages satisfy 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 the cases of the overlapping resolutions. The Silhouette Score is also the mean of the silhouette coefficients of all the cases of the overlapping resolutions. The outcome of each procedure is considered an approximation of the clustering stability and internal index of each algorithm.

A. Dataset of Unit Circle

The dataset of Unit Circle consists of approximately 150 noisy points, which are located on a unit circle. A unit circle is a circle of radius one and 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 [15], [19], [41], [45]. Fig. 3 shows the visualization of this 2-D dataset.

All the three algorithms, Mapper, F-Mapper, and SFCM, are used in order to analyze the dataset of Unit Circle. The detailed choice of the parameters for each algorithm is described in Table II. The parameters for the Mapper algorithm have been kept similar to those in the previous popular works. The density-based spatial clustering of applications with noise (DBSCAN) algorithm is set to default from the scikit-learn package in this experiment. The parameters for the F-Mapper 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 both the algorithms are colored by the same function, which is the mean of the coordinates of data points. Red color expresses the maximal value, and the blue one corresponds with the minimal value.

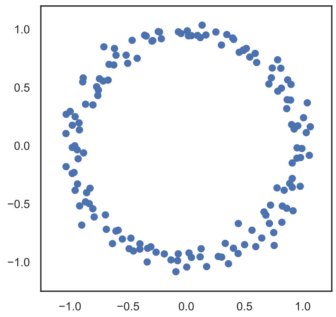


Fig. 3. Visual presentation of the dataset of Unit Circle. The x-axis and y-axis are corresponding to the first and second coordinates of the points of the dataset.

TABLE II
PARAMETER SETTINGS IN THE EXPERIMENT FOR THE DATASET OF UNIT CIRCLE

Mapper	***Lens***	***Number of intervals***	***Overlapping percentage***	***Clustering method***
	Sum	$N = 7$	$p = 50\%$	DBSCAN
F-Mapper	***Lens***	***Number of intervals***	***Overlapping threshold***	***Clustering method***
	Sum	$N = 7$	$\tau = 0.25$	DBSCAN
SFCM	***Number of clusters***		***Overlapping threshold***	
	$N = 12$		$\tau = 0.20$	

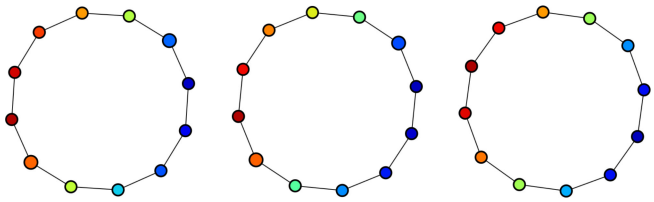


Fig. 4. Output comparison for all the algorithms on the dataset of Unit Circle. The images, from left to right, sequentially belong to the algorithms, Mapper, F-Mapper, and SFCM, respectively.

The experimental results of these algorithms on this dataset are shown in Fig. 4. The left image belongs to the Mapper algorithm, the center one belongs to the F-Mapper algorithm, and the right one belongs to the SFCM algorithm. These images are quite similar in terms of shape, structure, color, and size of nodes. This figure proves that the SFCM algorithm can create an output quite similar to that of both the Mapper and F-Mapper algorithms from a topological perspective.

In [41], the F-Mapper algorithm can produce the similar topological results as the Mapper algorithm can. Therefore, in this experiment, we have also evaluated the effectiveness of the SFCM algorithm not only with the Mapper algorithm but also with the F-Mapper algorithm through Matching Score and Silhouette Score.

All of the three algorithms, Mapper, F-Mapper, and SFCM, were run to change the overlapping parameters such that there was no change in the shape of the outputs with the other parameters fixed. We have the following results.

1) For the Mapper algorithm: The overlapping percentage p has varied from 42% to 50%.

TABLE III
MATCHING SCORE AND SILHOUETTE COEFFICIENT SCORE REPORT IN THE EXPERIMENT FOR THE DATASET OF UNIT CIRCLE

	Matching Score	Silhouette Score
Mapper	0.861	0.122
F-Mapper	0.878	0.147
SFCM	0.883	0.213

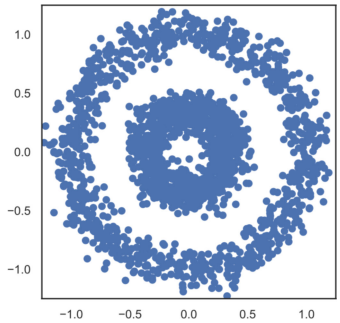


Fig. 5. Visual presentation of the dataset of Two Concentric Circles with Noise. The x-axis and y-axis are corresponding to the first and second coordinates of the points of the dataset.

2) For the F-Mapper algorithm: The overlapping threshold τ has been varied from 0.10 to 0.25 [41].
3) For the SFCM algorithm: The overlapping threshold τ has varied from 0.08 to 0.20.

For each algorithm, we obtain the following results.

1) The matching coefficient is calculated by using the method of k-fold cross validation for each case in which the overlapping parameters satisfy the nonchanged shape condition. In this experiment, k is set at 10 for subsampling the data, the same as the sampling ratio of 0.8 used in [65]. The Matching Score is the mean of the matching coefficients of all cases.
2) The silhouette coefficient is also calculated for each case where the overlapping parameters do not change the shape of the output of the algorithms. The Silhouette Score is the mean of the silhouette coefficients of all cases.

The Matching Scores and the Silhouette Score of the three algorithms are reported in Table III.

B. Dataset of Two Concentric Circles with Noise

The dataset of Two Concentric Circles with Noise has approximately 2000 noise points that create a large circle containing a smaller circle. This dataset is often used to visualize clustering and classification algorithms [69]. The visualization of this dataset is presented in Fig. 5 on the Euclidean plane.

The dataset of Two Concentric Circles with Noise is mined in turn with the three algorithms, Mapper, F-Mapper, and SFCM. Their respective parameters are described in Table IV. The parameters for the Mapper algorithm have been selected so that the output can detect the two circles' structure. The DBSCAN algorithm is also used with default settings from the scikit-learn package. The parameters for the F-Mapper and the SFCM algorithm are chosen to show that both of them can achieve the same results as the Mapper algorithm in terms of topology. The

TABLE IV
PARAMETER SETTINGS IN THE EXPERIMENT FOR THE DATASET OF TWO CONCENTRIC CIRCLES WITH NOISE

Mapper	***Lens***	***Number of intervals***	***Overlapping percentage***	***Clustering method***
	Sum	$N = 5$	$p = 5\%$	DBSCAN
F-Mapper	***Lens***	***Number of intervals***	***Overlapping threshold***	***Clustering method***
	Sum	$N = 5$	$\tau = 0.20$	DBSCAN
SFCM	***Number of clusters***		***Overlapping threshold***	
	$N = 12$		$\tau = 0.20$	

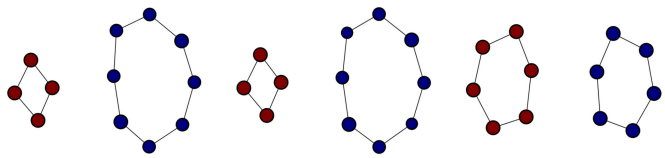


Fig. 6. Output comparison for all the algorithms on the dataset of Two Concentric Circles with Noise. The images, from left to right, sequentially belong to the algorithms Mapper, F-Mapper, and SFCM, respectively.

nodes in the graphs obtained by these algorithms are colored by the same function, which is the mean of the coordinates of data points. The colors range from red to blue expresses the colored values range from high to low.

Fig. 6 shows the results of the three algorithms obtained for this dataset. Three images, from left to right, sequentially belong to the algorithms Mapper, F-Mapper, and SFCM, respectively. They are almost similar in terms of shape, structure, color, and size of nodes. The images generated by two algorithms, Mapper and F-Mapper, consist of two loops corresponding to eight and four nodes, and the SFCM image consists of two loops that have 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 and F-Mapper algorithms in a topological perspective. Nevertheless, the most important thing in these pictures is that the data points are divided into two discrete classes with 100% accuracy for all the algorithms. This proves that the classification efficiency of the SFCM algorithm can reach equivalence in visualization with those of the Mapper and F-Mapper algorithms.

Similar to the previous experiments, we also evaluated the effectiveness of the SFCM algorithm not only with the Mapper algorithm but also with the F-Mapper algorithm through Matching Score and Silhouette Score. All 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. We have the following results.

1) For the Mapper algorithm: The overlapping percentage p has been altered from 3% to 7%.
2) For the F-Mapper algorithm: The overlapping threshold τ has been altered from 0.19 to 0.40.
3) For the SFCM algorithm: The overlapping threshold τ has been altered from 0.15 to 0.22.

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

TABLE V
MATCHING SCORE AND SILHOUETTE COEFFICIENT SCORE REPORT IN THE EXPERIMENT FOR THE DATASET OF TWO CONCENTRIC CIRCLES WITH NOISE

	Matching Score	Silhouette Score
Mapper	0.821	0.208
F-Mapper	0.867	0.200
SFCM	0.900	0.306

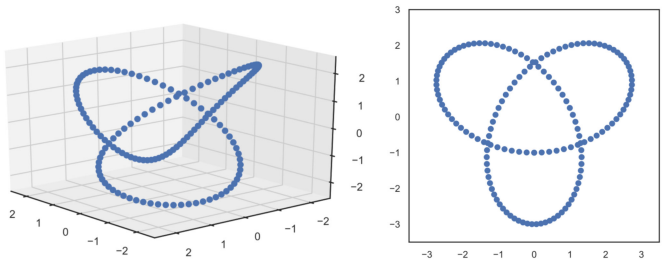


Fig. 7. Visual presentation of the dataset of 3-D Trefoil Knot. Left: In the $Oxyz$ space. Right: Projection in the Oxy plane.

TABLE VI
PARAMETER SETTINGS IN THE EXPERIMENT FOR THE DATASET OF 3D TREFOIL KNOT

Mapper	***Lens***	***Number of intervals***	***Overlapping percentage***	***Clustering method***
	l^2-Norm	$N = 2$	$p = 50\%$	DBSCAN
F-Mapper	***Lens***	***Number of intervals***	***Overlapping threshold***	***Clustering method***
	l^2-Norm	$N = 2$	$\tau = 0.40$	DBSCAN
SFCM	***Number of clusters***		***Overlapping threshold***	
	$N = 6$		$\tau = 0.25$	

For each algorithm, the Matching Score and Silhouette Score are also arranged in the same calculation as in the previous experiment. The scores of the three algorithms are reported in Table V.

C. Dataset of 3D Trefoil Knot

The dataset of 3-D Trefoil Knot has approximately 150 points that create a 3-D 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 approaches (e.g., t-distributed stochastic neighbor embedding) approaches [43]. The visualization of this advantage can be found in [43]. The visualization of this dataset is presented in Fig. 7 on the Euclidean space.

Now, we use the three algorithms, Mapper, F-Mapper, and SFCM, to process the dataset of 3-D Trefoil Knot. Table VI clearly expresses the choice of parameters corresponding to each algorithm. The parameters for the Mapper algorithm have been chosen so that its output can detect the circle structure as in the previous implementation [43]. The clustering algorithm used in the Mapper algorithm is l^2-Norm, a specific norm on a Euclidean vector space, that is strongly related with the Euclidean distance and equals the square root of the inner product of a vector with itself. The F-Mapper and SFCM algorithms are designed so that their outputs are the same as that of the Mapper algorithm in the view of topology. The nodes in the graphs of both the algorithms are colored by the same function, which is the height coordinate of data points. The high and low values are displayed in red and blue, respectively.

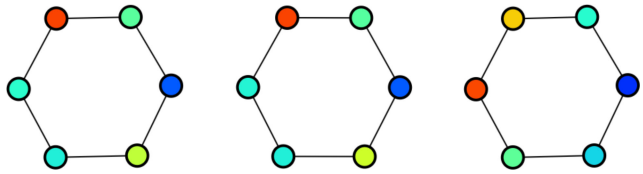


Fig. 8. Output comparison for all of algorithms on the dataset of 3D Trefoil Knot. The images, from left to right, sequentially belong to the algorithms Mapper, F-Mapper, and SFCM, respectively.

TABLE VII
MATCHING SCORE AND SILHOUETTE COEFFICIENT SCORE REPORT IN THE EXPERIMENT FOR THE DATASET OF 3D TREFOIL KNOT

	Matching Score	Silhouette Score
Mapper	0.891	0.017
F-Mapper	0.971	0.144
SFCM	0.925	0.205

The outputs of all the algorithms on this dataset are presented in Fig. 8. From left to right, the images belong to the Mapper, F-Mapper, and SFCM algorithms, respectively. These images are identical in shape, structure, and size of nodes. The color of nodes is the same for the outputs of both Mapper and F-Mapper algorithms, but it is a little different from that of the remaining algorithm. The reason for this difference is that the Mapper and F-Mapper algorithms use the lens function, but the SFCM algorithm does not. On the whole, the output of the SFCM algorithm is almost similar to those of the Mapper and F-Mapper algorithms with regard to the topological structure.

Similarly, the effectiveness of the SFCM algorithm was evaluated with both the Mapper and the F-Mapper algorithms through Matching Score and Silhouette Score. Once again, the overlapping parameters of all the algorithms were adjusted such that there was no change in the shape with the other parameters fixed. We have the following results.

1) For the Mapper algorithm: The overlapping percentage p has been fluctuated from 13% to 50%.
2) For the F-Mapper algorithm: The overlapping threshold τ has been fluctuated from 0.05 to 0.42.
3) For the SFCM algorithm: The overlapping threshold τ has been fluctuated from 0.21 to 0.30.

For each algorithm, similarly, the Matching Score and Silhouette Score are designed to calculate as same as in the previous experiments. These scores of the three algorithms are reported in Table VII.

D. Dataset of Reaven and Miller Diabetes

The dataset of Reaven and Miller Diabetes was one of the study results in the 1970s by Stanford University [47]. A total of 145 nonobese 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 dataset of Reaven and Miller Diabetes, which

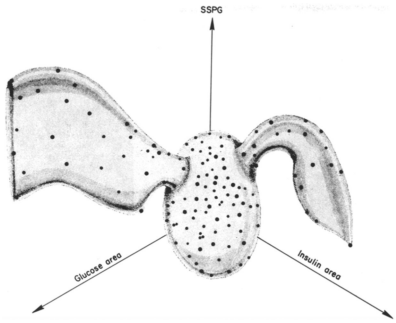


Fig. 9. Through the projection pursuit method, the dataset of Reaven and Miller Diabetes has a 3-D visual representation with a fat middle core and two floppy wings [70].

TABLE VIII
PARAMETER SETTINGS IN THE EXPERIMENT FOR THE DATASET OF THE REAVEN AND MILLER DIABETES

Mapper	***Lens***	***Number of intervals***	***Overlapping percentage***	***Clustering method***
	KDE	$N = 5$	$p = 50\%$	Single-Linkage
F-Mapper	***Lens***	***Number of intervals***	***Overlapping threshold***	***Clustering method***
	KDE	$N = 5$	$\tau = 0.06$	Single-Linkage
SFCM	***Number of clusters***		***Overlapping threshold***	
	$N = 10$		$\tau = 0.20$	

consisted 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 [70] visualized this dataset directly by the projection pursuit method and obtained a 3-D shape as shown in Fig. 9. This 3-D visual representation was interpreted as a boomerang with a fat middle core and two floppy wings [70]. The authors have indicated that the central core 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 the 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 VIII. The parameters for the Mapper algorithm have been kept the same as the well-known experiment in the original paper [15]. The lens used in this case is the kernel density estimate (KDE) function in statistics. Besides, the single-linkage clustering from the scikit-learn package with the default parameter setting is used in the clustering algorithm. The F-Mapper and the SFCM algorithms were exploited in the condition that 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 colors ranging from red to blue represent the colored values ranging from high to low.

Fig. 10 shows the outputs after processing by the algorithms on this diabetes dataset. From left to right, the images belong to the algorithms, Mapper, F-Mapper, and SFCM, respectively. In each algorithm, the central core that expresses the normal patients appears by red nodes. Both wings that express patients

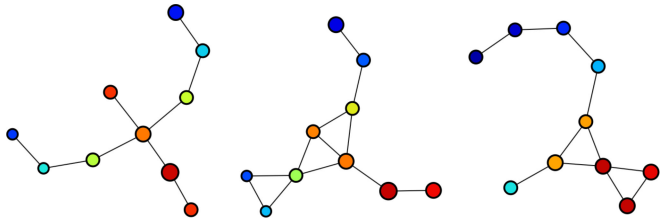


Fig. 10. Output comparison for all the algorithms on the dataset of Reaven and Miller Diabetes. The images, from left to right, sequentially belong to the algorithms Mapper, F-Mapper, and SFCM, respectively.

TABLE IX
MATCHING SCORE AND SILHOUETTE COEFFICIENT SCORE REPORT IN THE EXPERIMENT FOR THE DATASET OF THE REAVEN AND MILLER DIABETES

	Matching Score	**Silhouette Score**
Mapper	0.889	-0.032
F-Mapper	0.880	-0.028
SFCM	0.892	0.036

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 is not the case when the F-Mapper and SFCM algorithms are implemented. In another way, the connections between the nodes are different in the three algorithms, especially with regard to the positions of the left wings. The visualizations created by the F-Mapper 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 sparser than those in the others. There are some triangular forms in the outputs of the F-Mapper 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 lens is usually considered by the real-valued continuous function. Nevertheless, this connectivity occurs between three points to create the triangles in the image of the new later algorithms. As well as the F-Mapper algorithm, the overlap between clusters in the SFCM algorithm cannot necessarily be pairwise and consecutive.

We repeat the process of evaluating the clustering stability through the Matching Score and the internal index with Silhouette Score for the SFCM algorithm in relation to the two algorithms, Mapper and F-Mapper. Once again, all the algorithms changed the overlapping parameters such that there was no change in the shape of the outputs with the other parameters fixed. We have the following results.

1) For the Mapper algorithm: The overlapping percentage p has been altered from 41% to 50%.
2) For the F-Mapper algorithm: The overlapping threshold τ has been altered from 0.056 to 0.062 [41].
3) For the SFCM algorithm: The overlapping threshold τ has been altered from 0.20 to 0.24.

For each algorithm, similarly, the Matching Score and Silhouette Score are also carried out the same computation as in the previous experiments. The scores of the three algorithms are reported in Table IX.

TABLE X
PARAMETER SETTINGS AND RUN TIME REPORTS IN THE EXPERIMENTS FOR THE BIG DATASETS

Dataset	Dimension	Number of clusters	Overlapping threshold	Run time (s)
3D Lion	$(5000; 3)$	$N = 12$	$\tau = 0.180$	0.65
3D Cat	$(7206; 3)$	$N = 10$	$\tau = 0.217$	0.52
3D Horse	$(8430; 3)$	$N = 20$	$\tau = 0.117$	5.01
3D Road Network	$(434874; 4)$	$N = 12$	$\tau = 0.150$	119.76
Covertype	$(581012; 54)$	$N = 12$	$\tau = 0.160$	752.94

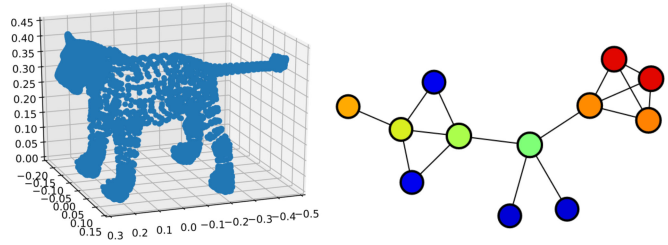


Fig. 11. 3-D visualization (left) and the SFCM visualization (right) of the 3D Lion dataset.

Overall, for the four real-world datasets, the experimental results focus on the output visualization from the topological standpoint, the clustering stability through the Matching Score, and the internal index with Silhouette Score for the three algorithms, including Mapper, F-Mapper, and SFCM. First, the results of the SFCM algorithm are quite similar to those of the Mapper and F-Mapper algorithms on the topological aspect in case the respective parameters are well-chosen. Secondly, the clustering stability based on the Matching Score and the internal index based on the Silhouette Score of this new proposed algorithm is better than those of Mapper and F-Mapper in most experimental cases.

To conclude this section, we applied the SFCM algorithm to visualize some large 3-D datasets, including Lion, Cat, Horse, Road Network, and Covertype, to prove the capability of the algorithm on big data. The datasets of Lion, Cat, and Horse were taken from the examples in the open-source codebase of the KeplerMapper [61], [62]. The 3D Road Network Dataset [71] and the Covertype Dataset [72] were taken from the UC Irvine Machine Learning Repository.

The choice of the parameters and the run time in the SFCM algorithm for each dataset are reported in Table X. The results of using the SFCM algorithm to mine these big datasets are shown in Figs. 11 –14.

The run-time results are perfectly acceptable for the high-dimensional datasets. Note that we do not compare the run-time between the new proposed algorithm and the original algorithms because their run-time result depends heavily on the time complexity of the lens function itself. The results are rather impressive for the 3-D animal datasets when nodes of the graph are colored by the height coordinate of data points. The red expresses maximum value and blue expresses minimum value. That is, 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 quite clearly on the graph: The feet are presented by blue nodes and the heads are presented by red nodes.

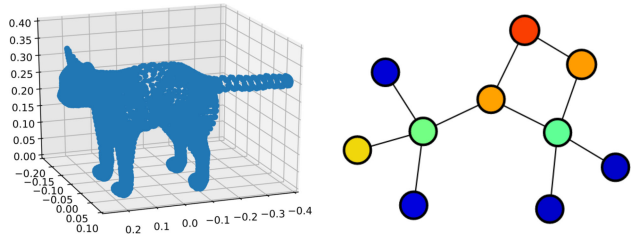


Fig. 12. 3-D visualization (left) and the SFCM visualization (right) of the 3D Cat dataset.

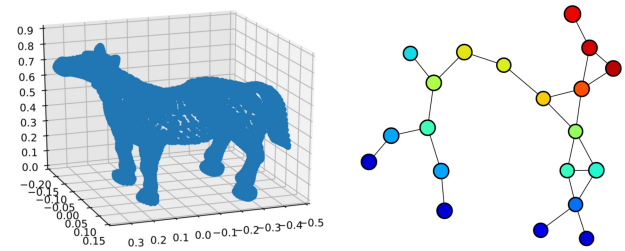


Fig. 13. 3-D visualization (left) and the SFCM visualization (right) of the 3D Horse dataset.

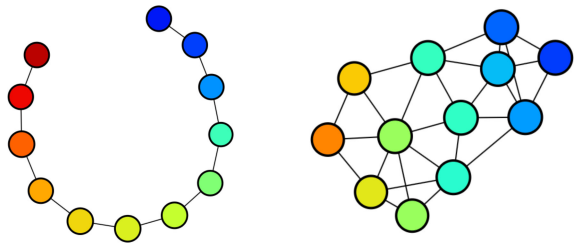


Fig. 14. SFCM visualization for the 3D Road Network Dataset (left) and the Covertype Dataset (right).

V. CONCLUSION

In this article, the SFCM algorithm was proposed as a fuzzy clustering algorithm endowed with the ability to create a simple, useful, and intuitive topological summary for high-dimensional datasets. It has been generated from the FCM algorithm by carefully combining the special ability for shape detection of the Mapper algorithm. On the one hand, the FCM algorithm was equipped with the outstanding features in simplifying and visualizing data with qualitative analysis. On the other hand, the SFCM algorithm also helped 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.

We demonstrated the effectiveness of the SFCM algorithm by experiments on the four real-world datasets. This method could produce results that are quite similar to those of the previous algorithms, Mapper and F-Mapper, on the topological aspect. Besides, the Matching Score that presented the clustering stability and the Silhouette Score that presented the internal

index of the new algorithm are better than those of the Mapper and F-Mapper in most experimental cases.

In a certain sense, the SFCM algorithm is considered as a special enhanced case of the F-Mapper algorithm [41]. The overlapping threshold in this special version, like the F-Mapper 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 toward 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 [57], [73]–[79]. These problems are expected to be addressed thoroughly in the near future to improve the algorithm with respect to both theory and applications.

REFERENCES

[1] J. Sheng, J. Amankwah-Amoah, and X. Wang, "A multidisciplinary perspective of big data in management research," *Int. J. Prod. Econ.*, vol. 191, pp. 97–112, 2017.

[2] D. Laney, "3D data management: Controlling data volume, velocity and variety," *META Group Res. Note*, vol. 6, no. 70, pp. 70–73, 2001.

[3] N. Oweis, S. Owais, W. George, M. Suliman, and V. Snasel, *A Survey on Big Data, Mining: (Tools, Techniques, Applications and Notable Uses)*, 2015.

[4] M. A. Khan, M. F. Uddin, and N. Gupta, "Seven V's of big data understanding big data to extract value," in *Proc. Zone 1 Conf. Amer. Soc. Eng. Educ.*, 2014, pp. 1–5.

[5] J. Andreu-Perez, C. C. Y. Poon, R. D. Merrifield, S. T. C. Wong, and G.-Z. Yang, "Big data for health," *IEEE J. Biomed. Healh Inform.*, vol. 19, no. 4, pp. 1193–1208, Jul. 2015.

[6] V. Snášel, J. Nowaková, F. Xhafa, and L. Barolli, "Geometrical and topological approaches to big data," *Futur. Gener. Comput. Syst.*, vol. 67, pp. 286–296, 2017.

[7] G. Carlsson, "Topology and data," *Bull. Amer. Math. Soc.*, vol. 46, no. 2, pp. 255–308, Jan. 2009.

[8] G. Carlsson, "Topological pattern recognition for point cloud data," *Acta Numer.*, vol. 23, pp. 289–368, 2014.

[9] A. Zomorodian, "Topological data analysis," *EPJ Data Sci.*, vol. 70, pp. 1–39, 2011.

[10] D. Beyer, *The Future of Machine Intelligence*. Newton, MA, USA: O'Reilly Media, Inc., 2016.

[11] F. Chazal and B. Michel, "An introduction to topological data analysis: Fundamental and practical aspects for data scientists," Oct. 2017, *arXiv:1710.04019*.

[12] R. Ghrist, "Barcodes: The persistent topology of data," *Bull. Amer. Math. Soc.*, vol. 45, pp. 61–75, 2008.

[13] S. Oudot, *Persistence Theory: From Quiver Representations to Data Analysis*. Providence, RI, USA: American Mathematical Society, 2016.

[14] N. Otter, M. A. Porter, U. Tillmann, P. Grindrod, and H. A. Harrington, "A roadmap for the computation of persistent homology," *EPJ Data Sci.*, vol. 6, no. 1, Aug. 2017, Art. no. 17.

[15] G. Singh, F. Memoli, and G. Carlsson, "Topological methods for the analysis of high dimensional data sets and 3D object recognition," in *Proc. Eurograph. Symp. Point-Based Graph.*, 2007, pp. 91–100.

[16] J. Milnor, M. Spivak, and R. Wells, *Morse Theory. (AM-51)*, vol. 51. Princeton, NJ, USA: Princeton University Press, 1969.

[17] Y. Yao *et al.*, "Topological methods for exploring low-density states in biomolecular folding pathways," *J. Chem. Phys.*, vol. 130, no. 14, Apr. 2009, Art. no. 144115.

[18] P. Y. Lum *et al.*, "Extracting insights from the shape of complex data using topology," *Sci. Rep.*, vol. 3, Feb. 2013, Art. no. 1236.

[19] K. Almgren, M. Kim, and J. Lee, "Extracting knowledge from the geometric shape of social network data using topological data analysis," *Entropy*, vol. 19, no. 7, pp. 1–17, 2017.

[20] M. Nicolau, A. J. Levine, and G. Carlsson, "Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival," *Proc. Nat. Acad. Sci.*, vol. 108, no. 17, pp. 7265–7270, 2011.

[21] J. L. Nielson *et al.*, "Topological data analysis for discovery in preclinical spinal cord injury and traumatic brain injury," *Nature Commun.*, vol. 6, Oct. 2015, Art. no. 8581.

[22] J. Rossi-deVries, V. Pedoia, M. A. Samaan, A. R. Ferguson, R. B. Souza, and S. Majumdar, "Using multidimensional topological data analysis to identify traits of hip osteoarthritis," *J. Magn. Reson. Imag.*, vol. 48, no. 4, pp. 1046–1058, Oct. 2018.

[23] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier Science, 2011.

[24] H. Edelsbrunner and J. Harer, *Computational Topology—An Introduction*. Providence, RI, USA: American Mathematical Society, 2010.

[25] L. A. Zadeh, "Fuzzy sets," *Inform. Control*, vol. 8, no. 3, pp. 338–353, 1965.

[26] V. B. Kuzmin, *Building Group Decisions in Spaces of Strict and Fuzzy Binary Relations*, Moscow: Nauka.

[27] F. Feng, H. Fujita, M. I. Ali, R. R. Yager, and X. Liu, "Another view on generalized intuitionistic fuzzy soft sets and related multiattribute decision making methods," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 3, pp. 474–488, Mar. 2019.

[28] C.-W. Lin, T. Li, P. Fournier Viger, and T.-P. Hong, "A fast algorithm for mining fuzzy frequent itemsets," *J. Intell. Fuzzy Syst.*, vol. 29, pp. 2373–2379, Oct. 2015.

[29] J. C.-W. Lin, T. Li, P. Fournier-Viger, T.-P. Hong, and J.-H. Su, "Fast algorithms for mining multiple fuzzy frequent itemsets," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2016, pp. 2113–2119.

[30] M. De Cock, U. Bodenhofer, and E. E. Kerre, "Modelling linguistic expressions using fuzzy relations," in *Proc. 6th Int. Conf. Soft Comput.*, 2000, pp. 353–360.

[31] F. Meng, J. Tang, and H. Fujita, "Linguistic intuitionistic fuzzy preference relations and their application to multi-criteria decision making," *Inform. Fusion*, vol. 46, pp. 77–90, 2019.

[32] D. Arotaritei and S. Mitra, "Web mining: A survey in the fuzzy framework," *Fuzzy Sets Syst.*, vol. 148, no. 1, pp. 5–19, 2004.

[33] C.-W. Lin and T.-P. Hong, "A survey of fuzzy web mining," *Wiley Interdiscip. Rev. Data Min. Knowl. Discovery*, vol. 3, no. 3, pp. 190–199, May 2013.

[34] L. R. Liang *et al.*, "FM-test: A fuzzy-set-theory-based approach to differential gene expression data analysis," *BMC Bioinformat.*, vol. 7, no. Suppl 4, pp. S7–S7, Dec. 2006.

[35] D. Xu, J. M. Keller, M. Popescu, and R. Bondugula, *Applications of Fuzzy Logic in Bioinformatics*, vol. 9. London, U.K.: Imperial College Press, 2008.

[36] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, no. 3, pp. 32–57, Jan. 1973.

[37] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1981.

[38] J. Yu, Q. Cheng, and H. Huang, "Analysis of the weighting exponent in the FCM," *IEEE Trans. Syst. Man Cybern. Part B*, vol. 34, no. 1, pp. 634–639, Feb. 2004.

[39] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 370–379, Aug. 1995.

[40] W. Pedrycz and J. V. de Oliveira, "A development of fuzzy encoding and decoding through fuzzy clustering," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 4, pp. 829–837, Apr. 2008.

[41] Q.-T. Bui, B. Vo, H.-A. N. Do, N. Q. V. Hung, and V. Snasel, "F-Mapper: A fuzzy mapper clustering algorithm," *Knowl.-Based Syst.*, vol. 189, 2020, Art. no. 105107.

[42] M. Saggar *et al.*, "Towards a new approach to reveal dynamical organization of the brain using topological data analysis," *Nature Commun.*, vol. 9, no. 1, 2018, Art. no. 1399.

[43] C. Geniesse, O. Sporns, G. Petri, and M. Saggar, "Generating dynamical neuroimaging spatiotemporal representations (DyNeuSR) using topological data analysis," *Netw. Neurosci.*, vol. 3, pp. 763–778, Apr. 2019.

[44] T. Wang, T. Johnson, J. Zhang, and K. Huang, *Topological Methods for Visualization and Analysis of High Dimensional Single-Cell RNA Sequencing Data*, 2019.

[45] K. Almgren, M. Kim, and J. Lee, "Mining social media data using topological data analysis," in *Proc. IEEE Int. Conf. Inf. Reuse Integr.*, 2017, pp. 144–153.

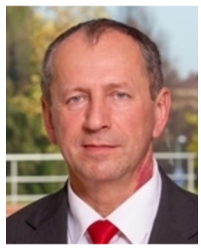
[46] A. Phinyomark, E. Ibáñez-Marcelo, and G. Petri, "Resting-state fMRI functional connectivity: Big data preprocessing pipelines and topological data analysis," *IEEE Trans. Big Data*, vol. 3, no. 4, pp. 415–428, Dec. 2017.
[47] T. K. Dey, F. Mémoli, and Y. Wang, "Multiscale mapper: Topological summarization via codomain covers," in *Proc. 27th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2016, pp. 997–1013.
[48] T. K. Dey, F. Mémoli, and Y. Wang, "Topological analysis of nerves, reeb spaces, mappers, and multiscale mappers," in *Proc. 33rd Int. Symp. Comput. Geometry*, 2017, pp. 36:1–36:16, doi: 10.4230/LIPIcs.SoCG.2017.36.
[49] R. Jeitziner, M. Carrière, J. Rougemont, S. Oudot, K. Hess, and C. Brisken, "Two-tier mapper: A user-independent clustering method for global gene expression analysis based on topology," Dec. 2017, *arXiv:1901.07410*.
[50] P. Dłotko, "Ball mapper: A shape summary for topological data analysis," Jan. 2019, *arXiv:1901.07410*.
[51] P. Dlotko, S. Rudkin, and W. Qiu, "An economic topology of the Brexit vote," Sep. 2019, *arXiv:1909.03490*.
[52] P. Dlotko, W. Qiu, and S. Rudkin, "Financial ratios and stock returns reappraised through a topological data analysis lens," Nov. 2019, *arXiv:1911.10297*.
[53] M. Carrière and S. Oudot, "Structure and stability of the one-dimensional mapper," *Found. Comput. Math.*, vol. 18, no. 6, pp. 1333–1396, 2018.
[54] M. Carrière, B. Michel, and S. Oudot, "Statistical analysis and parameter selection for mapper," *J. Mach. Learn. Res.*, vol. 19, no. 12, pp. 1–39, 2018.
[55] M. Hajij, B. Assiri, and P. Rosen, "Distributed mapper," 2017, *arXiv: 1712.03660*.
[56] M. Hajij, B. Wang, and P. Rosen, "MOG: Mapper on graphs for relationship preserving clustering," 2018, *arXiv:1804.11242*.
[57] J. Cyranka, D. Meyer, and A. Georges, "Mapper based classifier," Oct. 2019, *arXiv:1910.08103*.
[58] W. Huang, T. Hong, G. Lan, M. Chiang, and J. C. Lin, "Temporal-based fuzzy utility mining," *IEEE Access*, vol. 5, pp. 26639–26652, Nov. 2017.
[59] W.-M. Huang, T.-P. Hong, M.-C. Chiang, and J. C.-W. Lin, "Using multi-conditional minimum thresholds in temporal fuzzy utility mining," *Int. J. Comput. Intell. Syst.*, vol. 12, no. 2, pp. 613–626, 2019.
[60] L. Vendramin, M. Naldi, and R. Campello, "Fuzzy clustering algorithms and validity indices for distributed data," in *Proc. Partitional Clustering Algorithms*, 2015, pp. 147–192.
[61] N. Saul and H. J. van Veen, "MLWave/kepler-mapper: 186f," 2017. doi: 10.5281/ZENODO.1054444.
[62] H. Veen, N. Saul, D. Eargle, and S. Mangham, "Kepler Mapper: A flexible Python implementation of the Mapper algorithm," *J. Open Source Softw.*, vol. 4, Oct. 2019, Art. no. 1315.
[63] N. Saul and C. Tralie, "Scikit-TDA: Topological data analysis for Python." 2019. doi: 10.5281/zenodo.2533369.
[64] S. Ben-David, U. von Luxburg, and D. Pál, "A sober look at clustering stability," in *Proc. Int. Conf. Comput. Learn. Theory*, 2006, pp. 5–19.
[65] A. Ben-Hur, A. Elisseeff, and I. Guyon, "A stability based method for discovering structure in clustered data," *Pac. Symp. Biocomput.*, vol. 2002, pp. 6–17, Feb. 2002.
[66] U. von Luxburg, "Clustering stability: An overview," *Found. Trends Mach. Learn.*, vol. 2, no. 3, pp. 235–274, 2010.
[67] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987.
[68] R. J. G. B. Campello and E. R. Hruschka, "A fuzzy extension of the silhouette width criterion for cluster analysis," *Fuzzy Sets Syst.*, vol. 157, no. 21, pp. 2858–2875, 2006.
[69] F. Belchí, J. Brodzki, M. Burfitt, and M. Niranjan, "A numerical measure of the instability of Mapper-type algorithms," Jun. 2019, *arXiv:1906.01507*.
[70] G. M. Reaven and R. G. Miller, "An attempt to define the nature of chemical diabetes using a multidimensional analysis," *Diabetologia*, vol. 16, no. 1, pp. 17–24, 1979.
[71] C. Guo, Y. Ma, B. Yang, C. S. Jensen, and M. Kaul, "EcoMark: Evaluating models of vehicular environmental impact," in *Proc. 20th Int. Conf. Adv. Geograph. Inform. Syst.*, 2012, pp. 269–278.
[72] J. A. Blackard and D. J. Dean, "Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables," *Comput. Electron. Agric.*, vol. 24, no. 3, pp. 131–151, 1999.
[73] A. Kalyanaraman, M. Kamruzzaman, and B. Krishnamoorthy, "Interesting paths in the Mapper," 2017, *arXiv:1712.10197*.
[74] A. Brown, O. Bobrowski, E. Munch, and B. Wang, "Probabilistic convergence and stability of random Mapper graphs," Sep. 2019, *arXiv:1909.03488*.
[75] M. Carrière and B. Michel, "Approximation of Reeb spaces with Mappers and applications to stochastic filters," Dec. 2019, *arXiv:1912.10742*.
[76] Y. Shen, W. Pedrycz, Y. Chen, X. Wang, and A. Gacek, "Hyperplane division in fuzzy C-means: Clustering big data," *IEEE Trans. Fuzzy Syst.*, to be published. doi: 10.1109/TFUZZ.2019.2947231.
[77] S. Roh, S. Oh, W. Pedrycz, and Z. Fu, "Design of fuzzy ensemble architecture realized with the aid of FCM-based fuzzy partition and NN with weighted LSE estimation," *IEEE Trans. Fuzzy Syst.*, to be published. doi: 10.1109/TFUZZ.2019.2956903.
[78] T. Lei, X. Jia, Y. Zhang, S. Liu, H. Meng, and A. K. Nandi, "Superpixel-based fast fuzzy C-means clustering for color image segmentation," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 9, pp. 1753–1766, Sep. 2019.
[79] T. Feng, J. I. Davila, Y. Liu, S. Lin, S. Huang, and C. Wang, "Semi-supervised topological analysis for elucidating hidden structures in high-dimensional Transcriptome datasets," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, to be published. doi: 10.1109/TCBB.2019.2950657.

Quang-Thinh Bui received the B.S. degree in mathematics and the M.S. degree in geometry and topology from Ho Chi Minh City University of Education, Ho Chi Minh City, Vietnam, in 2009 and 2012, respectively. He is currently working toward the Ph.D. degree in applied mathematics at VSB-Technical University of Ostrava, Ostrava, Czech Republic.

From 2009 to 2019, he was a Lecturer with the Faculty of Education, Tien Giang University, Tien Giang, Vietnam, and a Researcher with Duy Tan University, Da Nang, Vietnam. His current research interest includes topology and data mining, specially applying topological techniques to extract shape information from datasets that are high-dimensional, incomplete, and noisy.

Bay Vo received the Ph.D. degree in computer science from the University of Science, Vietnam National University, Ho Chi Minh City, Vietnam, in 2011.

He is an Associate Professor and Dean of Faculty with Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam. He has published more than 140 papers in international journals/conferences. His current research interests include association rules, classification, mining in incremental database, distributed databases and privacy preserving in data mining, and soft computing.



Vaclav Snasel (Member, IEEE) received the M.Sc. degree in numerical mathematics from the Faculty of Science, Palacký University, Olomouc, Czech Republic, in 1981 and the Ph.D. degree in algebra and number theory from Masaryk University, Brno, Czech Republic, in 1991.

His research and development experience includes over 25 years in the Industry and Academia. He has authored or coauthored several refereed journal/conference papers and book chapters. He has published more than 400 papers (147 are recorded at Web of Science). He has supervised many Ph.D. students from Czech Republic, Jordan, Yemen, Slovakia, Ukraine, and Vietnam. Since 2001, he has been a Visiting Scientist with the Institute of Computer Science, Academy of Sciences of the Czech Republic. Since 2003, he has been the Vice-Dean for Research and Science with Faculty of Electrical Engineering and Computer Science, VSB-Technical University of Ostrava, Ostrava, Czech Republic. He has been a Full Professor since 2006. Before turning into a full time academic, he was working with an industrial company where he was involved in different industrial research and development projects for nearly eight years. He works in a multidisciplinary environment involving artificial intelligence, multidimensional data indexing, conceptual lattice, information retrieval, semantic web, knowledge management, data compression, machine intelligence, neural network, web intelligence, and data mining and applied to various real-world problems.

Dr. Snasel, besides being the Editor-in-Chief of two journals, also serves the editorial board of some reputed international journals. He is actively involved in the *International Conference on Computational Aspects of Social Networks (CASoN)*; *Computer Information Systems and Industrial Management (CISIM)*; *Evolutionary Techniques in Data Processing (ETID)* series of international conferences. He is a member of ACM, AMS, and SIAM.

Witold Pedrycz (Fellow, IEEE) received the M.Sc. degree in computer science, the Ph.D. degree in computer engineering, and the D.Sci. degree in system science from the Silesian University of Technology, Gliwice, Poland, in 1977, 1980, and 1984, respectively.

He is currently a Professor and Canada Research Chair (CRC) in Computational Intelligence with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada. He is also with the Systems Research Institute of the Polish Academy of Sciences, Warsaw, Poland. He has published numerous papers in these areas. His current h-index is 112 (Google Scholar) and 82 on the list *Top-h Scientists for Computer Science and Electronics*. He is also an author of 21 research monographs and edited volumes covering various aspects of computational intelligence, data mining, and software engineering. His main research interests involve computational intelligence, fuzzy modeling and granular computing, knowledge discovery and data science, pattern recognition, data science, knowledge-based neural networks, and control engineering.

Prof. Pedrycz is vigorously involved in editorial activities. He is an Editor-in-Chief for *Information Sciences* and *WIREs Data Mining and Knowledge Discovery* (Wiley) and Coeditor-in-Chief of *International Journal of Granular Computing* (Springer) and *Journal of Data Information and Management* (Springer). He serves on an Advisory Board of IEEE TRANSACTIONS ON FUZZY SYSTEMS and is a member of a number of editorial boards of international journals. In 2009, he was elected as a foreign member of the Polish Academy of Sciences. In 2012, he was elected as a Fellow of the Royal Society of Canada. In 2007, he was the recipient of a prestigious Norbert Wiener award from the IEEE Systems, Man, and Cybernetics Society. He is a recipient of the IEEE Canada Computer Engineering Medal, Cajastur Prize for Soft Computing from the European Centre for Soft Computing, Killam Prize, Fuzzy Pioneer Award from the IEEE Computational Intelligence Society, and 2019 Meritorious Service Award from the IEEE Systems Man and Cybernetics Society.

Tzung-Pei Hong (Senior Member, IEEE) received the B.S. degree in chemical engineering from National Taiwan University, Taipei, Taiwan, in 1985, and the Ph.D. degree in computer science and information engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 1992.

He has served with the Department of Computer Science, Chung-Hua Polytechnic Institute, Hsinchu, Taiwan, from 1992 to 1994, and with the Department of Information Management, I-Shou University, Kaohsiung, Taiwan, from 1994 to 2001. He was in charge of the whole computerization and library planning for National University of Kaohsiung, Taiwan, in Preparation from 1997 to 2000. He served as the First Director of the Library and Computer Center, National University of Kaohsiung from 2000 to 2001, as the Dean of Academic Affairs from 2003 to 2006, as the Administrative Vice President from 2007 to 2008, and as the Academic Vice President in 2010. He is currently a distinguished and chair Professor with the Department of Computer Science and Information Engineering and with the Department of Electrical Engineering, and the Director with AI Research Center, National University of Kaohsiung. He is also a Joint Professor with the Department of Computer Science and Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan. He has published more than 600 research papers in international/national journals and conferences and has planned more than 50 information systems. His current research interests include knowledge engineering, data mining, soft computing, management information systems, and www applications.

Dr. Hong was the recipient of the first national flexible wage award from Ministry of Education in Taiwan. He is also a board member of more than 40 journals and the program committee member of more than 900 conferences.

Ngoc Thanh Nguyen (Senior Member, IEEE) received the M.Sc. degree in information systems from the Wroclaw University of Technology, Wroclaw, Poland, in 1986, the Ph.D. degree in computer science from the Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland in 1989, and the D.Sci. degree in computer science from the Wroclaw University of Technology, Warsaw, in 2002.

He is currently a Full Professor with the Wroclaw University of Science and Technology and the Head of Applied Informatics Department. He is also the Honorary Chair of the Scientific Board with Nguyen Tat Thanh University, Ho Chi Minh City, Vietnam. He has authored or coauthored five monographs and more than 350 journal and conference papers. He has given 22 plenary and keynote speeches for international conferences and more than 40 invited lectures in many countries. His current research interests include collective intelligence, knowledge integration methods, inconsistent knowledge processing, and multi-agent systems.

Prof. Nguyen has edited more than 30 special issues in international journals, 52 books, and 42 conference proceedings. He serves as an Editor-in-Chief for the *International Journal of Information and Telecommunication* (Taylor&Francis), the *Transactions on Computational Collective Intelligence* (Springer), and *Vietnam Journal of Computer Science* (World Scientific). He is also an Associate Editor for several prestigious international journals, among others, IEEE TRANSACTIONS ON SMC: SYSTEMS, *Journal of Intelligent and Fuzzy Systems*, and *Applied Intelligence*. He was a General Chair or Program Chair of more than 40 international conferences. He serves as a member of the Council of Scientific Excellence of Poland, a member of Committee on Informatics of the Polish Academy of Sciences, an Expert of National Center of Research and Development and European Commission in evaluating research projects in several programs like Marie Sklodowska-Curie Individual Fellowships, FET, and EUREKA. In 2009, he was granted the title of Distinguished Scientist of ACM. He was also a Distinguished Visitor of the IEEE and a Distinguished Speaker of ACM. He also serves as the Chair for IEEE SMC Technical Committee on Computational Collective Intelligence.

Mu-Yen Chen received the Ph.D. degree in information management from National Chiao-Tung University, Hsinchu, Taiwan, in 2006.

He is a Associate Professor of Department of Engineering Science with the National Cheng Kung University, Tainan, Taiwan. His current research interests include artificial intelligence, soft computing, bio-inspired computing, data mining, deep learning, context-awareness, machine learning, and healthcare, with more than 100 publications in these areas.

Prof. Chen has served as an Associate Editor for *Journal of Information Processing Systems* and *International Journal of Social and Humanistic Computing*. He is an editorial board member of several SCI journals. In addition, he has also coedited 12 special issues in international journals, such as *Computers in Human Behavior*, *Applied Soft Computing*, *Soft Computing*, *Information Fusion*, *Neurocomputing*, *Journal of Medical and Biological Engineering*, *The Electronic Library*, *Library High Tech*, etc.