

Improving fuzzy clustering algorithm for probability density functions and applying in image recognition

Dinh Phamtoan^a and Tai Vovan^{b,*}

^a*Faculty of Engineering Van Lang University*

^b*College of Natural Science, Can Tho University, Can Tho City, Vietnam*

Abstract. This study introduces a measure called coefficient of within-cluster proximity (CWP) to evaluate the similarity of probability density functions (DFs) within clusters. After surveying the under and upper, and the computational problems of CWP, a fuzzy clustering algorithm for DFs is proposed. This algorithm can determine the suitable number of clusters and find the probability for each DF to belong to specific cluster. The convergence of the algorithm is considered in theory and illustrated by the numerical examples. The algorithm is applied to image recognition. The results show strong advantages of it in comparison to other algorithms. They also indicate the potential of the proposed approach in application to the data of different types.

Keywords: Automatic algorithm, density function, fuzzy cluster analysis, image recognition

1. Introduction

Cluster analysis is intended to separate the initial data into sub-groups or “clusters” in such a way that the elements in the same clusters are more similar to each other than the elements from other clusters based on any criteria (Thao & Tai, 2017). It is a crucial development of pattern recognition and an interesting topic of multidimensional statistics. Cluster analysis has a basis role in data mining as well as data analysis used in many areas. Therefore, it has been concerned by many statisticians (Chicco et al., 2003; Austin et al., 2005; Ciszak, 2008).

Normally, the elements for clustering are the discrete elements and the probability density functions (DFs). For these objects, there are two main approaches for clustering. They are hierarchical and non-hierarchical methods. Clustering of discrete elements (CDE) has been studied and widely used in many fields. Clustering for density function (CDF) has been developed in recent years from requirements of some applications. Since the DF can represent the information in an object, it has advantages in analysing for the complex data (Thao & Tai, 2017). For clustering, CDF shows the outstanding benefit when compared with CDE in many cases (Chen & Hung, 2015; Tai & Thao, 2018). As a result, it has been preferred in some recent studies. Goh and Vidal (2008) proposed the effective algorithm in computing because it reduced the number of dimensions based on the unit Hilbert sphere. Montanari and Calò (2013) introduced a method for CDF based on estimating non-parametric of DFs, and used the mixture models for hyper-spherical data. Chen and Hung (2015) proposed a simple algorithm that could analyse the automatic clusters for DFs. Thao and Tai (2017) contributed three different algorithms for fuzzy hierarchical and non-hierarchical methods. Tai and Thao (2018) proposed the non-fuzzy clustering algorithm for DFs by a new criterion in evaluating the similarity of elements. Recently, Diem et al. (2018a) proposed a fast and efficient approach for CDF by applying k -medoids scheme. In addition, another works of Diem et al. (2018b) solved CDF based on

*Corresponding author: Tai Vovan, College of Natural Science, Can Tho University, Can Tho City, Vietnam. E-mail: vvtai@ctu.edu.vn.

differential evolution for the first time and gained some interesting result. Nevertheless, these works did not have the algorithm to determine the suitable number of clusters. The number of clusters was often given as a parameter in the clustering process. This is an important disadvantage of the above algorithms. Furthermore, those methods do not give the best result when they perform for data which have much overlap as images.

The other aspect, how to evaluate the similarity of DFs is also an important problem. Until now, several distance notions have proposed for the separation of DFs. For example, Matusita (1967) proposed a measure based on the weighted product of DFs. However, it was very complex in computing and inefficient in clustering (Tai, 2017). In 2008, Pham-Gia et al. (2008) proposed the L^1 -distance of DFs that was improved from the separated measure of Glick (1973). However, it was only used for classification problem, not for cluster analysis. Inheriting from the work of Pham-Gia et al. (2008), Tai and Pham-Gia (2010) proposed a measure for DFs called the cluster width. Although this measure was used as a criterion to build CDF, it has shown the disadvantage with DFs that have higher overlap. This algorithm did not propose the method to determine the suitable number of clusters. The work of Chen and Hung (2015) overcame this drawback, and contributed an automatic clustering algorithm for DFs. In spite of these advantages, the method of Chen and Hung (2015) was still encouraged with DFs to have the high degree of overlap. Currently, Tai and Thao (2018) proposed a new measure to cluster for DFs. However, this measure did not be applied to the fuzzy cluster analysis. For CDE, the degree of “goodness” has been measured by a lot of indexes such as S-index, F-index, Dunn index, and Xie-Beni index (Dunn, 1974; Pal & Bezdek, 1995; Xie & Beni, 2012). Although above indexes have many contributions in the cluster construction, they are only computed after clusters have been established. Therefore, the problem finds the best among n given strategies, we need to do all of these n methods, which leads to the complexity of the whole procedure. Moreover, the above indexes only evaluate the goodness of general strategy and cannot evaluate the goodness of each established cluster. Tai and Thao (2019) proposed the similar coefficient of the cluster of discrete elements. In order to fill in the mentioned research gap, this article proposes a criterion called the coefficient of within-cluster proximity (CWP) for DFs. CWP not only can be used as a criterion to build CDF, but it also can be considered as a valid measure to evaluate established clusters qualities.

In cluster analysis, the crisp and fuzzy algorithms are the main approaches. In the crisp clustering, each element only belongs to a cluster with probability as 1. Therefore, some boundary elements may not be evaluated precisely. In fuzzy clustering, each element can be assigned to one or more clusters with different probabilities. With this feature, the fuzzy clustering is more advantageous than the crisp one. Fuzzy cluster analysis has been extensively studied for CDE, but it is limited for CDF (Tai & Thao, 2019). The important results about fuzzy clustering of CDF have presented in Chen and Hung (2015), Thao and Tai (2017). Since these studies used the normal distance as the criterion to evaluate the similarity of DFs, they had the disadvantage in applying.

Nowadays, in technological development, clustering analysis for images is very important thanks to its applications for different fields. There are several methods to classify the images into different groups. Almost all these methods have taken the input data based on the extracted features from the image such as colour, texture, co-occurrence matrix, and shape. Among them, the studies about texture and colour are preferred. Setia et al. (2006) used the co-occurrence matrices of local relational features to classify images, whereas Eleyan et al. (2011) introduced a new approach to recognize the face based on gray level co-occurrence matrices. Guo et al. (2005) segmented the images by HRRS method based on the unsupervised texture. These studies only performed with input data to be the discrete elements. Clustering for images based on DFs has not been given much attention. The significant contributions for this application were presented by Chen and Hung (2015) and Thao and Tai (2017). It is noticed that the above studies only focus on texture and shape from the features of the image. In addition, cluster analysis for images is usually used as a numerical example in these studies, not the major purpose. These days, in state-of-the-art studies, DFs is just estimated in one-dimensional space. For three-dimensional spaces, it is still restricted (Tai et al., 2019). We don't find the documents to compare the effectiveness of clustering for images with the extracted DFs in one-dimension and three-dimensions. The applications for images also show the reasonableness and the advantages of the proposed algorithm in comparison with the existing ones.

The rest of this article is structured as follows. Section 2 presents the definition the coefficient of within-cluster proximity, the bounds, and the problem to compute this measure. The proposed algorithm and its convergence are given in Section 3. Section 4 applies the proposed algorithm in recognizing images. Section 5 is conclusion. Appendix of the article gives the proofs of Theorems.

2. Coefficient of within-cluster proximity

2.1. The definitions

Let $f_1(x), f_2(x), \dots, f_k(x)$ be DFs on $R^n, n \geq 1$ and $f_{\max}(x) = \max\{f_1(x), f_2(x), \dots, f_k(x)\}$. Then, we have the following definitions:

Definition 1. The coefficient of within-cluster proximity (CWP) is defined by

$$c(f_1(x), f_2(x), \dots, f_k(x)) = \frac{k}{k-1} \left[1 - \frac{1}{k} \int_{R^n} f_{\max}(x) dx \right]. \quad (1)$$

The CWP of the cluster with only one DF is defined as 1.

For $k = 2$, Eq. (1) becomes:

$$c(f_1(x), f_2(x)) = 2 - \int_{R^n} f_{\max}(x) dx = 2 - \|f_1(x), f_2(x)\|_1 - \int_{R^n} f_{\min}(x) dx, \quad (2)$$

where $\|f_i(x), f_j(x)\|_1 = \int_{R^n} |f_i(x) - f_j(x)| dx$ and $f_{\min}(x) = \min\{f_1(x), f_2(x)\}$.

The larger for the value of CWP is, the higher for the similarity degree of the DFs is.

Definition 2. Let $g, g_1, g_2, \dots, g_{k_1}, f_1, f_2, \dots, f_{k_2}$ be the DFs. Then, we have the following definitions:

$$c[(g) \cup \{f_1, f_2, \dots, f_{k_2}\}] = \frac{k_2 + 1}{k_2} \left[1 - \frac{1}{k_2 + 1} \int_{R^n} \max(g, f_1, f_2, \dots, f_{k_2}) dx \right].$$

$$c[\{f_1, f_2, \dots, f_{k_1}\} \cup \{g_1, g_2, \dots, g_{k_2}\}] = \frac{k_1 + k_2}{k_1 + k_2 - 1} \left[1 - \frac{1}{k_1 + k_2} \int_{R^n} \max\{f_1, \dots, f_{k_1}, g_1, \dots, g_{k_2}\} dx \right].$$

Definition 3. Given the set of DFs $F = \{f_1(x), f_2(x), \dots, f_k(x)\}, k \geq 2$. If F is analysed to m clusters then the cluster probability matrix of F is defined $U = [\mu_{ij}]_{m \times k}$, with $\mu_{ij} \in [0, 1]$ is probability of the j^{th} element joining the i^{th} cluster.

In non-fuzzy cluster analysis, $\mu_{ij} = 1$ when the j^{th} element belongs to the i^{th} cluster and $\mu_{ij} = 0$ in the opposite cases.

Definition 4. Given the set of DFs $F = \{f_1(x), f_2(x), \dots, f_k(x)\}$, the prototype DF of F is defined as follows:

$$f_F(x) = \frac{\sum_{j=1}^k (\mu_{Fj})^m f_j(x)}{\sum_{j=1}^k (\mu_{Fj})^m}, \quad (3)$$

where μ_{Fj} is the probability for $f_j(x)$ belong to F , and m is positive interger called fuzziness parameter.

From Eq. (3), we can prove that $f_F(x) \geq 0$ with all of x and $\int_{R^n} f_F(x) dx = 1$. Therefore, one prototype element of a cluster is also a DF.

2.2. Bound of CWP

Theorem 1. Given the set of DFs: $F = \{f_1(x), f_2(x), \dots, f_k(x)\}, k \geq 3$, the following properties are satisfied:

$$\text{a) } 1 - \frac{1}{k(k-1)} \sum_i \sum_j \|f_i(x), f_j(x)\|_1 \leq c(f_1(x), \dots, f_k(x)) \leq 1 - \frac{1}{2(k-1)} \max_{i < j} \{\|f_i(x), f_j(x)\|_1\}, \quad (4)$$

$$\text{b) } 0 \leq c(f_1, f_2, \dots, f_k) \leq 1, \quad (5)$$

where $\lambda_{1,2,\dots,k} = \int_{R^n} \min\{f_1(x), f_2(x), \dots, f_k(x)\} dx$ is the measure of the overlap of F .

Proof The proof of this theorem is presented in *Appendix 1*.

2.3. Computing CWP

To compute CWP, we need to find the $\int_{R^n} f_{\max}(x) dx$. If $f_{\max}(x)$ is found then $\int_{R^n} f_{\max}(x) dx$ can be determined

easily. The specific expression for $f_{\max}(x)$ in some special cases has been found in Tai and Pham-Gia (2010), and Thao and Tai (2017). However, the general expression for all cases is a complex problem that still has not been found yet.

Given k DFs $\{f_i(x), i = 1, 2, \dots, k\}$. The maximum function $f_{\max}(x) = \max\{f_1(x), f_2(x), \dots, f_k(x)\}$ is determined in the following two cases.

i) For one-dimension

- Step 1. Solve the equations $f_i(x) - f_j(x) = 0, i = 1, 2, \dots, k - 1, j = i + 1, \dots, k$, to find all roots.
 Step 2. With root x_{lm} of equation $f_l(x) - f_m(x) = 0$, compare value $f_l(x_{lm})$ with all the values of $f_j(x_{lm}), j \neq l, m$. If there exists $p \neq l, m$ such that $f_p(x_{lm}) > f_l(x_{lm})$ then we delete x_{lm} and keep x_{lm} for otherwise. Arrange the kept roots in order from smallest to largest, then we have a roots set $B = \{x_1, x_2, \dots, x_h\}$.
 Step 3. Give $i = 1, 2, \dots, k; j = 1, 2, \dots, h$, function $f_{\max}(x)$ is determined by the following principles:
 If $\max\{f_1(x_1 - \varepsilon_1), f_2(x_1 - \varepsilon_1), \dots, f_k(x_1 - \varepsilon_1)\} = f_i(x_1 - \varepsilon_1)$
 then $f_{\max}(x) = f_i(x)$ for $x \in (-\infty, x_1)$.
 If $\max\{f_1(x_j + \varepsilon_2), f_2(x_j + \varepsilon_2), \dots, f_k(x_j + \varepsilon_2)\} = f_i(x_j + \varepsilon_2), j = 1, 2, \dots, h - 1$
 then $f_{\max}(x) = f_i(x)$ if $x \in (x_i, x_{i+1})$.
 If $\max\{f_1(x_h - \varepsilon_3), f_2(x_h - \varepsilon_3), \dots, f_k(x_h - \varepsilon_3)\} = f_i(x_h - \varepsilon_3)$
 then $f_{\max}(x) = f_i(x)$ if $x \in (x_h, +\infty)$.
 In the above algorithm, $\varepsilon_1, \varepsilon_2, \varepsilon_3$ are the positive constants so that:

$$x_1 + \varepsilon_1 < x_2, x_h - \varepsilon_3 > x_{h-1}, x_i - \varepsilon_2 < x_{i-1} \text{ and } x_i + \varepsilon_2 < x_{i+1}.$$

From this algorithm, we have established a Matlab procedure to find the $f_{\max}(x)$. When $f_{\max}(x)$ is determined, we will easily calculate CWP by Eq. (1).

ii) For multi-dimensions

In case of multi-dimensions, it should be very complicated to obtain the closed expression for $f_{\max}(x)$. The difficulty comes from the various forms of the intersection space curves between the surfaces of DFs. This problem has been interested in by several authors (Tai & Pham-Gia, 2010; Thao & Tai, 2017; Tai & Thao, 2018) who have attempted to find the function $f_{\max}(x)$. However, it has only been established for some cases of bivariate normal distribution. In this research, we do not find the expression of $f_{\max}(x)$. We compute CWP by taking integration of $f_{\max}(x)$ by quasi Monte-Carlo method. An algorithm for doing calculations has been constructed, and a corresponding Matlab procedure is also established.

3. The proposed algorithm

3.1. The algorithm

Let $F = \{f_1(x), f_2(x), \dots, f_k(x)\}$, be the set of DFs and $Fv^{(t)} = \{fv_1^{(t)}, fv_2^{(t)}, \dots, fv_k^{(t)}\}$ be the sequences on k representative DFs of clusters in the iteration t . The proposed algorithm has the following steps.

- Step 1. – Set $Fv = \{fv_1, fv_2, \dots, fv_k\} = \{f_1, f_2, \dots, f_k\}$ be the initial representing DFs.
 – Set ε is very small positive number.
 Step 2. Update the representing DFs by the formula:

$$fv_i^{(t)} = \frac{\sum_{j=1}^k K(fv_i, fv_j)fv_j}{\sum_{j=1}^k K(fv_i, fv_j)}, i = 1, \dots, k. \quad (6)$$

where

$$K(fv_i, fv_j) = \begin{cases} \exp\left(\frac{c(fv_i, fv_j) - 1}{\lambda}\right) & \text{if } c(fv_i, fv_j) \geq c_s, \\ 0 & \text{if } c(fv_i, fv_j) < c_s, \end{cases} \quad (7)$$

with $\lambda = \frac{1-c_s}{r}$, $c_s = \frac{1}{\binom{k}{2}} \sum_{i < j} c(fv_i, fv_j)$, and r is constant.

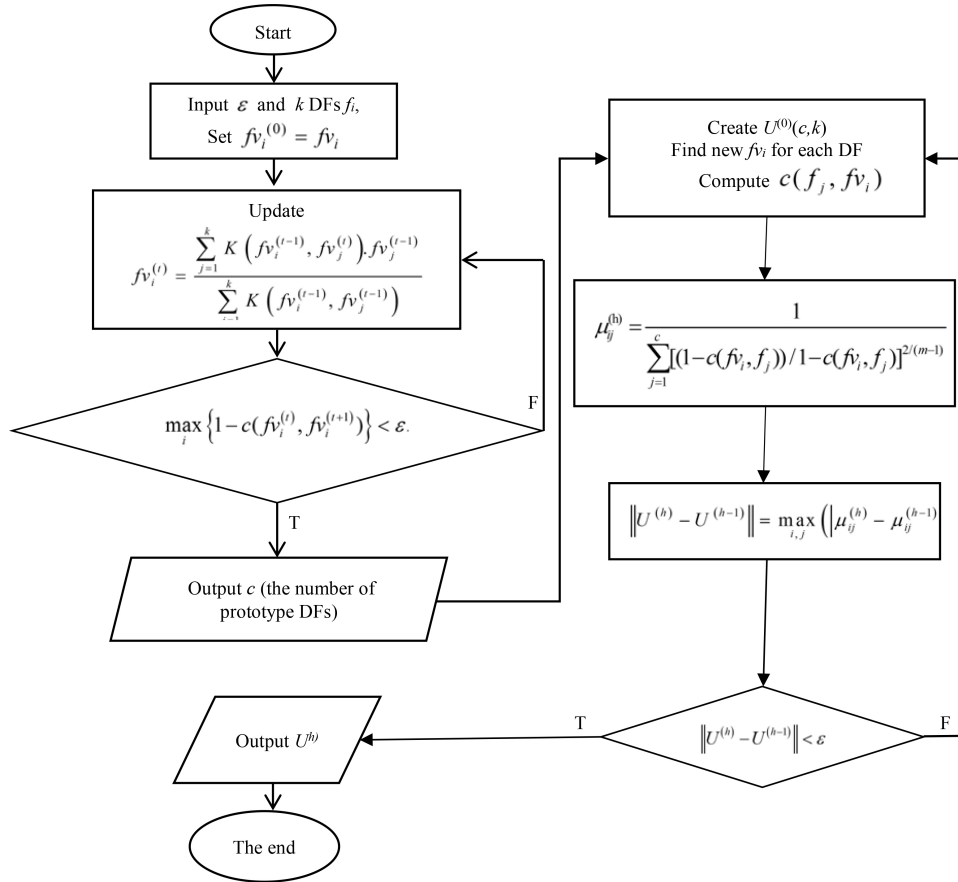


Fig. 1. The diagram of the proposed algorithm.

Step 3. Repeat Step 2 until $\max_i \{1 - c(f_{v_i}^{(t)}, f_{v_i}^{(t-1)})\} < \varepsilon$.

After Step 3 stops, the DFs in the same cluster will converge to the representative DF. If there are c representing DFs then we have c clusters.

Step 4. Let c be the number of the prototype DFs when Step 3 stops. Finding prototype DFs f_{v_i} for each cluster by Eq. (3) and computing distance between every DF to f_{v_i} by Eq. (1).

Step 5. – Create the initial cluster probability matrix $U^{(0)}$ with c rows and k columns. In this matrix, probability is chosen randomly.

– Find prototype $f_{v_i}, i = 1, 2, \dots, c$ for each cluster by Eq. (3) and compute CWP between every DF and f_{v_i} by Eq. (1).

Step 6. Update the new cluster probability matrix U^{new} by the following rule:

$$\mu_{ij}^{new} = \begin{cases} \frac{1}{\sum_{j=1}^k [(1 - c(f_{v_i}, f_j))/1 - c(f_{v_j}, f_i)]^{2/(m-1)}} & \text{if } c(f_j, f_{v_i}) < 1, i = 1, 2, \dots, c \\ 0 & \text{otherwise} \end{cases}$$

Step 7. Calculate $\|U^{(h)} - U^{(h-1)}\| = \max_{i,j} (|\mu_{ij}^{(h)} - \mu_{ij}^{(h-1)}|)$.

Step 8. Repeat Step 6 and Step 7 h times until the following condition is satisfied:

$$\|U^{(h)} - U^{(h-1)}\| < \varepsilon.$$

After Step 8 ends, we have a matrix with c rows and k columns.

Diagram of the proposed method is presented in Fig. 1.

3.2. Some problems of the proposed algorithm

In the proposed algorithm, we need to notice the following problems.

- (i) ε is a really small number, and it is chosen arbitrarily. The smaller ε is, the more iterations and computers time are taken. In this article, we choose $\varepsilon = 0.0001$.
- (ii) m is the fuzziness parameter, when $m = 1$, the fuzzy clustering becomes the non-fuzzy clustering. When $m \rightarrow \infty$, the partition becomes completely fuzzy with $\mu_{ij} = 1/c$. Generally, it is difficult to determine the optimal m . Fadili et al. (2001), Bora and Gupta (2014), and Yu et al. (2004) proposed the methods to determine the value of m . In many applications, the meshing method of Thao and Tai (2017) is often used. This research also uses above method to determine the value of m . It often has the value from 2 to 5.
- (iii) The value of λ affects the number of clusters. When $\lambda \rightarrow 0$ each of the elements is their own clusters, and when $\lambda \rightarrow \infty$, the data have only one cluster. The value of λ depends on r . Although there are discussions about choosing r (Chen & Hung, 2015), the optimal choice has not found. By experimenting with many data, we take $r = 5$ for the proposed algorithm.
- (iv) In practice, data usually contain discrete elements, so we firstly have to estimate the DFs to apply the proposed algorithm. There are many methods to solve this problem. This research uses the kernel function method, the most popular one in practice. It has the following form:

$$\hat{f}(x) = \frac{1}{N h_1 h_2 \dots h_n} \sum_{i=1}^N \prod_{j=1}^n f_j \left(\frac{x_j - x_{ij}}{h_j} \right), \quad (8)$$

where $x_j, j = 1, 2, \dots, n$ are variables, $x_{ij}, i = 1, 2, \dots, N$ are i th data of j th variable, h_j is bandwidth parameter for the j th variable, and $f_j(\cdot)$ is kernel function of j th variable.

For Eq. (8), the choice of smoothing parameter and the type of kernel function are important. Although some authors such as Tai (2017), Thao and Tai (2017) had many discussions about this problem, the optimal choice still has not been found yet. In this article, the smoothing parameter is chosen by Thao and Tai (2017), and the kernel function is the Gaussian kernel function (Thao & Tai, 2017).

- (v) We have established the complete Matlab procedure for the proposed algorithm. This algorithm can extract the features of images based on their colours, find the suitable number of clusters, determine the elements in each cluster, and give the probability belonging to clusters of each element at the same time. It has been applied quickly and effectively for examples and applications of this article.

3.3. The convergence of the proposed algorithm

The proposed algorithm has two phases. Phase 1 contains Step 1, Step 2, and Step 3, which provides the suitable number c of clusters. The number of clusters continues to be used in Phase 2 (from Step 4 to Step 8) for finding the clustering solution. At the end of Phase 2, the final clustering result is obtained. Actually, the steps in Phase 2 are done in the similar way as in c -means fuzzy clustering algorithm for discrete elements in which its convergence proved in many documents such as (Hathaway & Bezdek, 1988). Although we apply to another measure in clustering for DFs, the convergence of the proposed algorithm is still preserved. Therefore, in this section, we focus on the proof of the convergence of the Phase 1 by the Theorem 2.

Theorem 2. Let $Fv^{(t)} = \{fv_1^{(t)}, fv_2^{(t)}, \dots, fv_k^{(t)}\}$ be the sequences on DFs of c clusters in the iteration t . For each fv_i of Fv , there exists at least one j , such that $\lim_{t \rightarrow \infty} fv_i^{(t)} = fv_j$.

Proof The proof of this theorem is presented in Appendix 2.

3.4. Illustration for the algorithm

Given 100 univariate normal DFs $\{f_i(x)\}, i = 1, 2, \dots, 100$, with all variance equal to 1 and the following means:

- 50 DFs with $\mu_i = 0.1i, i = 1, 2, \dots, 50$.
- 20 DFs with $\mu_i = 10 + 0.1i, i = 1, 2, \dots, 20$.
- 30 DFs with $\mu_i = 30 + 0.1i, i = 1, 2, \dots, 30$.

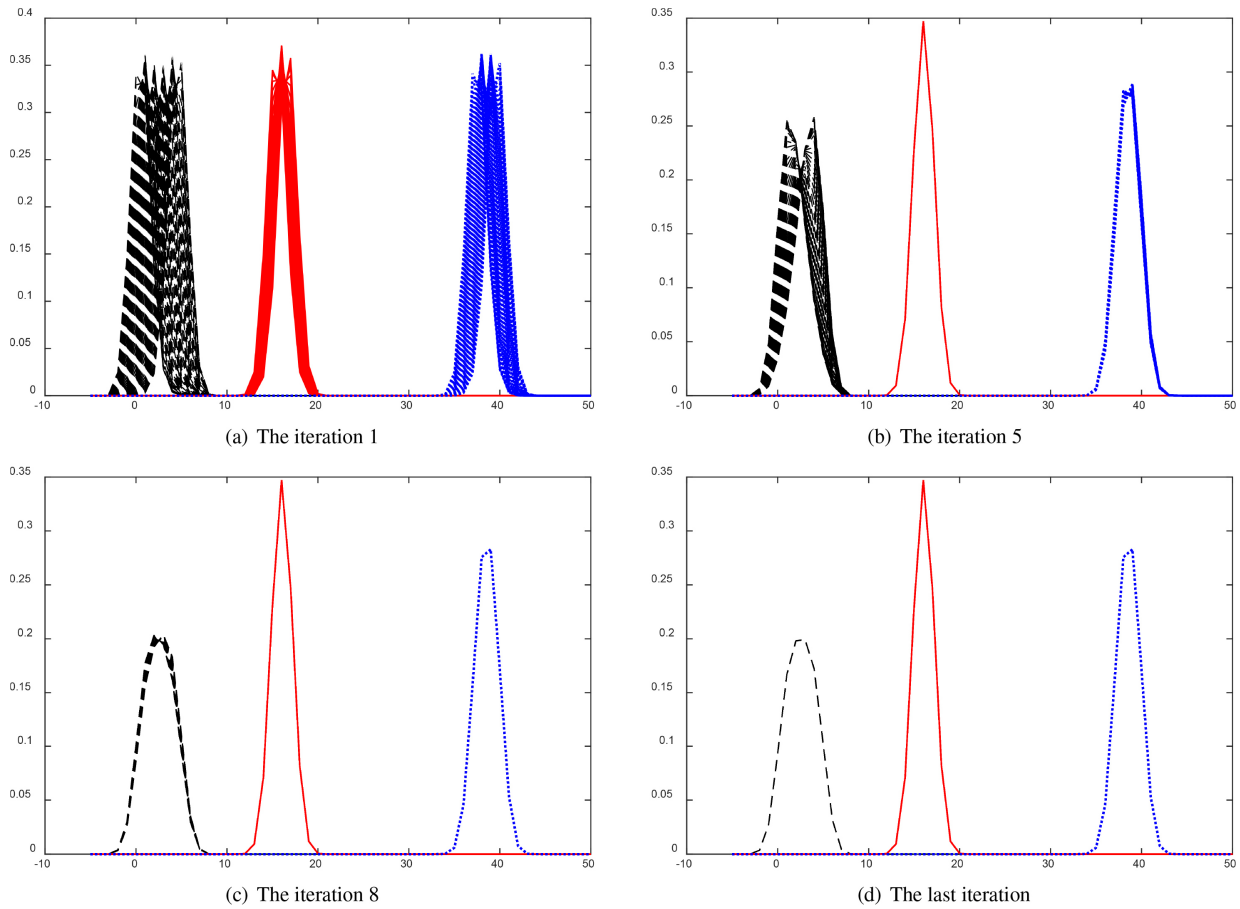


Fig. 2. The convergence of Phase 1 for 100 DFs.

Perform Phase 1, after 10 iterations (see Fig. 2), we have 3 clusters:

$$C_1 = \{f_1, f_2, \dots, f_{50}\}, C_2 = \{f_{51}, f_{52}, \dots, f_{70}\}, C_3 = \{f_{71}, f_{72}, \dots, f_{100}\}.$$

Continuing Phase 2 with 15 iterations, we receive the final probability partition matrix as follows:

$$\begin{bmatrix} 0.249 & 0.244 & \dots & 0.776 & 0.812 & \dots & 0.150 & 0.142 \\ 0.506 & 0.517 & \dots & 0.112 & 0.094 & \dots & 0.149 & 0.141 \\ 0.245 & 0.239 & \dots & 0.112 & 0.094 & \dots & 0.701 & 0.717 \end{bmatrix}.$$

These probabilities are shown in Fig. 3.

In short, the proposed algorithm has given the good result in this data. The number of clusters has determined suitably and the probability of each DF belongs to its right cluster.

4. Applying in image recognition

4.1. Problem of extracting color features for images

i) Using the G scale

The gray level co-occurrence matrix (GLCM) for an image with size $M \times N$ and G gray level is a matrix P of size $G \times G$. Each element $p_{d\theta}(i, j)$ of P presents the probability of the occurrence of intensity i and j with the distance d

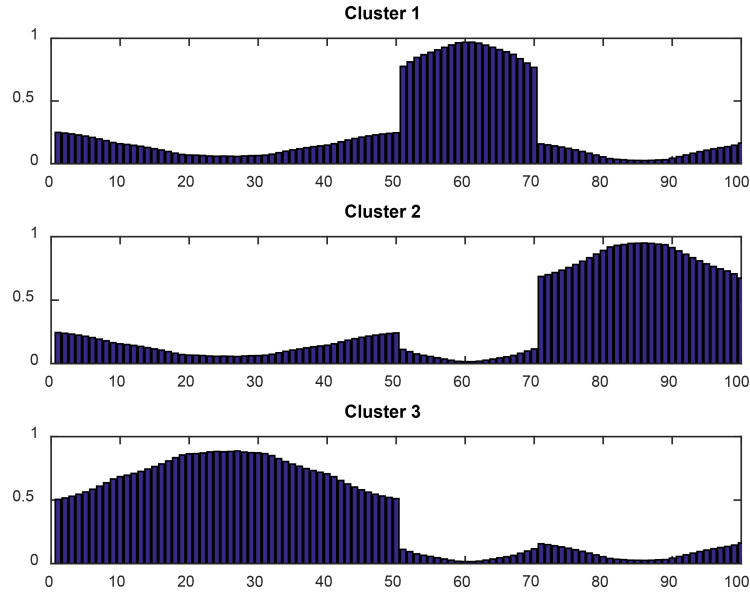


Fig. 3. Probability of 100 normal DFs belonging to 3 clusters.

and the orientation angle θ . The formula to compute $p_{d\theta}(i, j)$ is given by Eq. (9).

$$p_{d\theta}(i, j) = \left\{ \begin{array}{l} ((r, c), (r', c')) \in M \times N \\ \left. \begin{array}{l} d = \|(r, c), (r', c')\| \\ \theta = \Theta((r, c), (r', c')) \\ I(r, c) = i, I(r', c') = j \end{array} \right\} \right\}, \quad (9)$$

where $((r, c), (r', c')) = \max\{|r - r'|, |c - c'|\}$.

After calculating GLCM for each image, using the Eq. (8) to estimate density function. It means that each image is now represented by a DF of one variable.

ii) Using the RGB scale

The image is extracted by three colors (R: Red, G: Gray, B: Blue) that each color is also performed similar as i). In this case, each image is represented by a DF of three variables.

4.2. Numerical examples

In this section, two numerical examples are performed to apply the proposed algorithm in clustering of images. For these examples, the images are examined on both G and RGB scales. Example 1 realizes the CURET images with three main types of structures (see <http://www.uv.uio.no/~tranden/brodatz.html>). Example 2 includes 2 image sets about persons and flowers considered in Tai et al. (2019). Example 1 presents in detail the steps of the proposed algorithm, and compares it with the existing algorithms. Example 2 only compares the proposed algorithm with others. The compared existing approaches are non-herarchical of Tai and Pham-Gia (2010), automatic clustering of Chen and Hung (2015), and modified genetic algorithm (GA-CDF) of Tai et al. (2017). It is known that the above algorithms require the number of clusters in clustering process, except the algorithm of Chen and Hung (2015). Therefore, we use the number of clusters obtained from the proposed algorithm to compare the clustering result of methods. The indexes used to compare are mis-clustering rate (MR), Alternative Silhouette width criterion (ASWC), PBM (acronym constituted of the initials of the names of its authors, Pakhira, Bandyopadhyay and Maulik), and adjusted rand index (ARI). Detail about these indexes is given in Vendramin et al. (2010).

For the parameters ASWC, PBM, and ARI, the larger these parameters are, the better the algorithm are. However, the value of MR has the opposite meaning. Moreover, to confirm the stability as well as the accuracy, each algorithm would be executed in 30 independent iterations and the result will be appreciated by an average value.

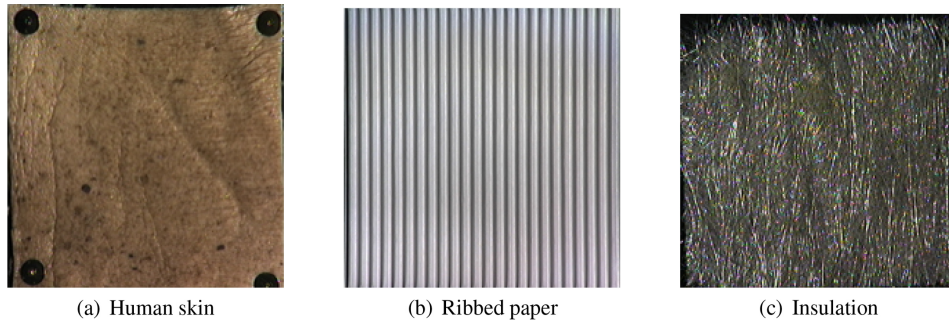


Fig. 4. Three original texture samples in CURET database.

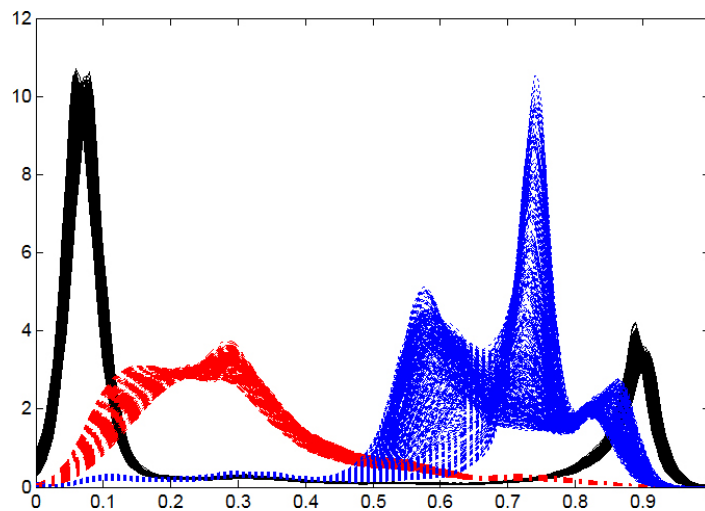


Fig. 5. The estimated DFs based on G scale of 218 images.

Example 1. This example considers 218 images with size 256×256 pixels in the CURET database. It is separated into three groups including human skin, ribbed paper, and insulation with the numbers as 57, 77 and 84, respectively. Some images of the three groups are presented in Fig. 4.

The DFs are estimated based on the G and RGB scales of the image pixels. In case of G scale, the DFs is demonstrated by Fig. 5.

Perform Phase 1, both G and RGB scales give the number of clusters as 3. For example, the 5th and final iterations are shown in Figs 6 and 7.

Execute Phase 2 of the proposed algorithm, we have the probability partition matrixes in cases of G and RGB scales as follows:

– Grayscale:

$$\begin{bmatrix} 0.260 & 0.524 & \dots & 0.295 & 0.223 & \dots & 0.715 & 0.716 \\ 0.615 & 0.321 & \dots & 0.227 & 0.233 & \dots & 0.160 & 0.159 \\ 0.125 & 0.115 & \dots & 0.478 & 0.544 & \dots & 0.125 & 0.125 \end{bmatrix}.$$

– RGB scale:

$$\begin{bmatrix} 0.210 & 0.264 & \dots & 0.447 & 0.513 & \dots & 0.222 & 0.187 \\ 0.499 & 0.335 & \dots & 0.257 & 0.243 & \dots & 0.239 & 0.197 \\ 0.291 & 0.401 & \dots & 0.296 & 0.244 & \dots & 0.539 & 0.616 \end{bmatrix}.$$

These probabilities for two cases are shown by Figs 8 and 9, respectively.

In both the above two matrixes, the probability each image assigned to its right cluster is fairly high. Comparing the proposed algorithm to the existing ones, we have Table 1.

Table 1
Comparing the algorithms performed in the Grayscale of CURET database

Scale	Method	MR	ASWC	PBM	ARI
G	Proposed	0.02	1.45	1.01	0.95
	Non-hierarchical	0.33	0.36	0.12	0.16
	GA-CDF	0.60	0.46	0.36	0.34
	Automatic clustering	0.61	0.48	0.42	0.46
RGB	Proposed	0.01	1.35	0.86	0.96
	Non-hierarchical	0.67	0.29	0.05	0.06
	GA-CDF	0.56	0.49	0.28	0.36
	Automatic clustering	0.05	1.21	0.72	0.87

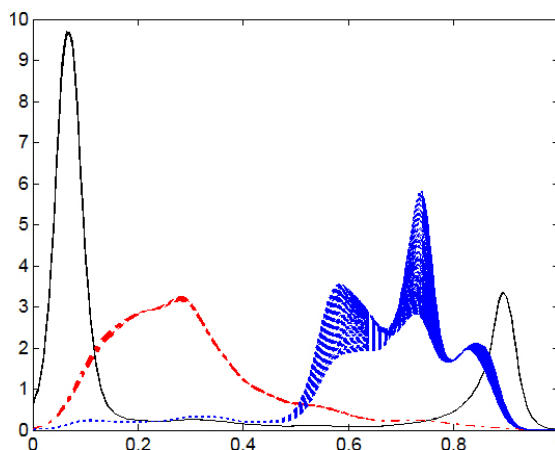


Fig. 6. The 5th iteration for CURET database in G scale.

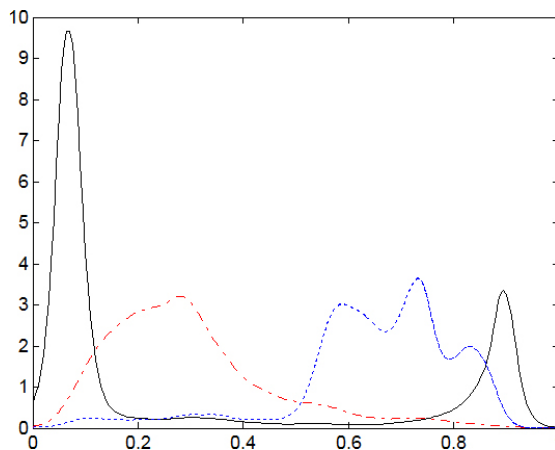


Fig. 7. The final iteration for CURET database in G scale.

In short, the proposed algorithm has given the accurate result about the number of clusters with this image data. The probability belonging to its right cluster is also suitable. In addition, Table 1 shows that the proposed algorithm has the best result in comparing to the existing ones for all the considered parameters.

Example 2. This example uses two data sets considered in Tai et al. (2019). Data 1 contains the images about persons divided by two groups: sensitive and insensitive. The number of groups is 49 and 50, respectively. Data 2 clusters of flowers separated into three groups: Passion, Gazania, and Lotus. The number of groups 192, 76, and 251, respectively. Some image samples of Data 1 and Data 2 are given in Figs 10 and 11, respectively.

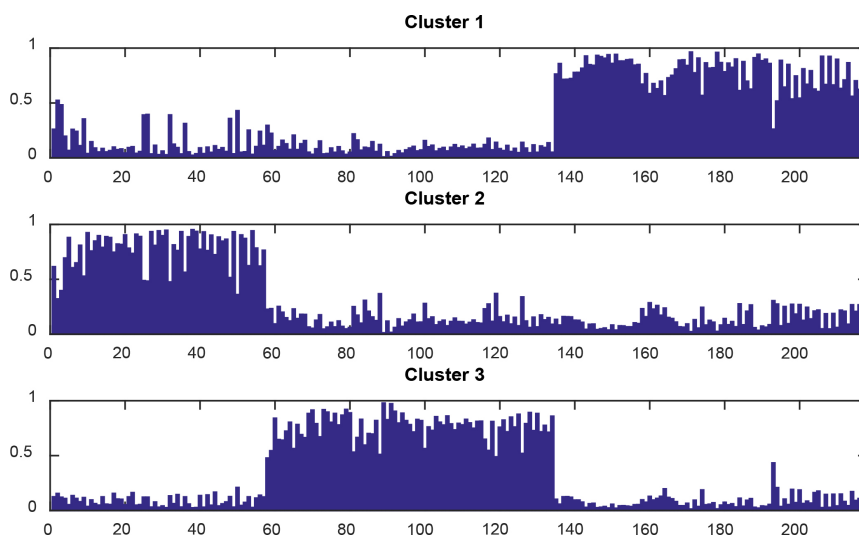


Fig. 8. Probability of each image belonging to three clusters in G scale.

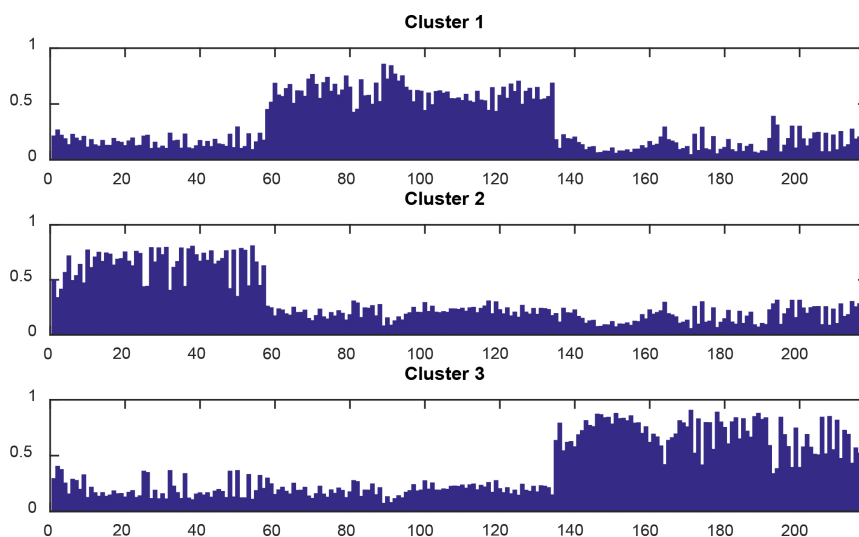


Fig. 9. Probability of each image belonging to three clusters in RGB scale.



(a) Sensitive images

(b) Insensitive images

Fig. 10. Sample images of two groups for Data 1.

Table 2
Comparing the result of algorithms or Data 1

Scale	Method	MR	ASWC	PBM	ARI
G	Proposed	0.00	1.67	1.02	1.00
	Non-hierarchical	0.32	0.77	0.16	0.15
	GA-CDF	0.15	1.18	0.78	0.86
	Automatic clustering	0.44	1.35	0.65	0.87
RGB	Proposed	0.00	1.42	0.83	1.00
	Non-hierarchical	0.34	0.81	0.17	0.09
	GA-CDF	0.15	1.2	0.63	0.90
	Automatic clustering	0.15	1.12	0.63	0.90

Table 3
Comparing the result of algorithms for Data 2

Scale	Method	MR	ASWC	PBM	ARI
G	Proposed	0.14	1.37	0.16	0.67
	Non-hierarchical	0.48	0.74	0.04	0.09
	GA-CDF	0.52	1.13	0.06	0.09
	Automatic clustering	0.59	1.17	0.03	0.01
RGB	Proposed	0.14	2.20	0.56	0.68
	Non-hierarchical	0.48	0.78	0.06	0.08
	GA-CDF	0.51	1.89	0.46	0.05
	Automatic clustering	0.59	1.03	0.08	0.01



Fig. 11. Sample images of three groups for Data 2.

Implementing Phase 1 of the proposed algorithm, we obtain the number of clusters for Data 1 and Data 2 are 2 and 3, respectively. These are suitable results for two data sets. The algorithm of Chen and Hung (2015) gives 2 clusters for both two data sets. It means that the number of clusters is only suitable for Data 1, and it is not for Data 2. Because other algorithms do not have the approaches to determine the number of clusters, we use the number of clusters from the proposed algorithm to compare the clustering process. They are shown in Table 2 (Data 1) and Table 3 (Data 2).

Although the extracted DFs from two data sets have high overlap, the proposed algorithm still obtains good results in determining the suitable number of clusters, and finding the probability to belong to clusters of images. Table 2 and Table 3 show that the proposed algorithm has the best result in comparing to the existing ones.

5. Conclusion

This paper proposes the fuzzy clustering algorithm for probability density functions with several improvements. Firstly, a measure called CWP is proposed for evaluating the within-cluster proximity. Secondly, defining bound of CWP as well as calculating CWP in real problem are also established. Then, the specific steps of the proposed algorithm based on CWP are presented. This algorithm can determine the suitable number of clusters, the elements in each cluster, and give the probability belonging to clusters of each DF. The convergence of the proposed algorithm is also considered in theory and checked by the numerical examples. This algorithm can perform effectively by the established Matlab procedure. Finally, the practical application of the proposed algorithm is considered in the image recognition problem. Various types of images with different sizes and quantities in many fields have shown that the

proposed algorithm outperforms the existing ones. This research can extend for practical applications in other fields such as security, medical, and economics. This would be one of our indispensable researches.

Acknowledgments

The authors would like to thank the anonymous reviewers for their careful reading and insightful suggestions of the manuscript. We wish to thank Lien Nguyenthi, the lecturer of Van Lang University, who has the helps to complete this article.

References

- Austin, S. B., Steven, J. M., Brisa, N., & Sanchez, A. (2005). Clustering of fast-food restaurants around schools: A novel application of spatial statistics to the study of food environments. *American Journal of Public Health*, 95(9), 1575-1581.
- Bora, D. J., & Anil, K. G. (2014). Impact of exponent parameter value for the partition matrix on the performance of fuzzy c means algorithm. *ArXiv Preprint ArXiv*, 406.4007.
- Chen, J. H., & Hung, W. L. (2015). An automatic clustering algorithm for probability density functions. *Journal of Statistical Computation and Simulation*, 85(15), 3047-3063.
- Chicco, G. Napoli, R., & Piglion, F. (2003). Application of clustering algorithms and self organising maps to classify electricity customers. *Power Tech Conference Proceedings, 2003 IEEE Bologna*, 1, 1-7.
- Ciszak, L. (2008). Application of clustering and association methods in data cleaning. *Proceedings of the International Multiconference on Computer Science and Information Technology*, 97-103.
- Diem, H. K., Trung, V. D. Trung, N. T. Tai, V. V., & Thao, N. T. (2018a). A differential evolution-based clustering for probability density functions. *IEEE Access*, 6380568, 1-19.
- Diem, H. K., Tai, V. V., & Thao, N. T. (2018b). Clustering for probability density functions by new-medoids method. *Scientific Programming*, ID 2764016, 1-7.
- Dunn, I. J. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics* 4, 95-104.
- Eleyan, A., Demirel, H. (2011). Co-occurrence matrix and its statistical features as a new approach for face recognition. *Turk J Elec Eng Comp Sci*, 19, 97-107.
- Fadili, M. J., Ruan, S., Bloyet, D., & Mazoyer, B. (2001). On the number of clusters and the fuzziness index for unsupervised FCA application to BOLD fMRI time series. *Med Image Anal*, 5(1), 55-67.
- Glick, N. (1973). Separation and probability of correct classification among two or more distributions. *Annals of the Institute of Statistical Mathematics*, 25(1), 373-382.
- Goh, A., & Vidal, R. (2008). Unsupervised riemannian clustering of probability density functions. *Machine Learning and Knowledge Discovery in Databases*, 1, 15-29.
- Guo, D., Atluri, V., & Adam, N. (2005). Texture-based remote-sensing image segmentation. *IEEE Int Conf Multimed*, 1472-1475.
- Hathaway, R. J., & Bezdek, J. C. (1988). Recent convergence results for the fuzzy c-means clustering algorithms. *Journal of Classification*, 5(2), 237-247.
- Matusita, K. (1967). On the notion of affinity of several distributions and some of its applications. *Annals of the Institute of Statistical Mathematics*, 19(1), 181-192.
- Montanari, A. and Calò, D. G. (2013). Model-based clustering of probability density functions. *Advances in Data Analysis and Classification*, 7(3), 301-319.
- Pal, N. R., & Bezdek, J. C. (1995). On cluster validity for the fuzzy c-means model. *IEEE Trans Fuzzy System*, 3, 370-379.
- Pham-Gia, T., Turkkan, N., & Tai, V. V. (2008). Statistical discrimination analysis using the maximum function. *Communications in Statistics - Simulation and Computation*, 37(2), 320-336.
- Setia, L., Teynor, A., Halawani, A., & Burkhardt, H. (2006). Image classification using cluster cooccurrence matrices of local relational features. *Proc 8th ACM Int*, 173-182.
- Tai, V. V., & Pham-Gia, T. (2010). Clustering Probability Distributions. *Journal of Applied Statistics*, 37(11), 1891-1910.
- Tai, V. V. (2017). L^1 - distance and classification problem by Bayesian method. *Journal of Applied Statistics*, 44(3), 385-401
- Tai, V. V., Trung, N. T., Trung, V. D., Huu, H. H., and Thao, N. T. (2017). Modified genetic algorithm-based clustering for probability density functions. *Journal of Statistical Computation and Simulation*, 87(10), 1964-1979.
- Tai, V. V., & Thao, N. T. (2018). Cluster similar of cluster for probability density functions. *Communication in Statistics – Theory and Methods*, 47(8), 1792-1811.
- Tai, V. V., & Thao, N. T. (2019). Similar coefficient of cluster for discrete elements. *Sankhya B, The Indian Journal of Statistics*, 80(1), 19-36.
- Tai, V. V., Dinh, P. T., and Dung, T. T. T. (2019). Automatic genetic algorithm in clustering for discrete elements. *Computation Statistics-Simulation and Communication*. doi: 10.1080/036109118.2019.158830.
- Thao, N. T., & Tai, V. V. (2017). Fuzzy clustering of probability density functions. *Journal of Applied Statistics*, 44(4), 583-601.
- Vendramin, L., Ricardo, J. G., Campello, B., & Eduardo, R. H. (2010). Relative clustering validity criteria: A comparative overview. *Statistical Analysis and Data Mining*, 3(4), 9-25.
- Yu, J., Qiansheng, C., & Houkuan, H. (2004). Analysis of the weighting exponent in the FCM. *IEEE Transactions on Cybernetics*, 34(1), 634-649.
- Xie, X. L., & Beni, G. (1991). A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13, 841-847.