Can Tho University Journal of Science
website: sj.ctu.edu.vn

DOI: 10.22144/ctu.jen.2017.047

Fuzzy cluster analysis for probability density functions based on width criterion

Vo Van Tai, Pham Toan Dinh and Nguyen Hoang Yen
College of Natural Science, Can Tho University, Vietnam

Article info.

Received 08 May 2017
Revised 28 Jul 2017
Accepted 31 Oct 2017

Keywords

Clustering, density, distance, fuzzy, width

ABSTRACT

Based on the cluster width of probability density functions, the algorithms have been established for fuzzy cluster analysis and for determining the suitable number of cluster. In addition, determining the width of cluster for two and more than two probability density functions has been also considered by Matlab produces. *The numerical examples in both synthetic and real data are given not only to illustrate the reasonable of proposed algorithms and programs but also to show their advantages in comparing with existing ones.*

Cited as: Tai, V.V., Dinh, P.T. and Yen, N.H., 2017. Fuzzy cluster analysis for probability density functions based on width criterion. Can Tho University Journal of Science. 7: 37-44.

1 INTRODUCTION

Clustering can arrange unknown large data into smaller groups so that elements in each group having some similar properties. This is an initial sorting step to get basic information from the data before implementing deeper analysis. In global trend, storing, extracting and analyzing data play an important role and have an influence on the development of theories and application of different science subjects. For this reason, cluster analysis problem has been interested by many statisticians so far. Cluster analysis can be applied for identifying images of satellite and medicine in discovering computer virus and in many internet problems, etc. It is possible to build clusters for discrete elements (CDE) (Fukunaga, 1990, Defays, 1997) and clusters for probability density functions (CPD) (Tai and Pham-Gia, 2010). CPD has consideration to the distribution of data in performing, so it is appreciated more advantages than CDE in many cases, especially in data mining (Tai and Pham-Gia, 2010; Chen and Hung, 2015).

In both cases, the most important problem is to seek a suitable criterion for building clusters and evaluating their quality. For CDE, the distance is the main criterion to perform. There are Euclidean distance, Chebyshev distance, city block distance for two elements and min distance, max distance, mean distance for two groups (Web, 2003). For two probability density functions (pdfs), some types of distances have been widely used such as the L^p- distance, the Bhattacharya distance, the Divergence distance, the Helinger distance (Defays, 1997; Webb, 2003). For more than two pdfs, some general and specific measures have been also introduced. They were the affinity of Matusita and Toussaint (Matusita, 1967; Toussaint, 1971) and the separated measure of Glick (1973). However, there is no above measures used as criterion in CPD. Because these measures are defined by integrating the weighted product of pdfs, the calculations in performing is complicated. In addition, the visualization of these measures is not obvious. According to Pham-Gia and Turkkan (2006), Pham-Gia *et al.* (2008), Montanari and Calò (2013), the criterion to perform CPD has not been much studied yet. Based on the distance of two pdfs, Goh and Vidal (2008) had initial contribution for CPD by a new algorithm, and then Montanari and Calò (2013) has improved this algorithm. These contributions did not have the algorithm for determining the appropriate number of clusters. From the concept about the separated measure of Glick (1973), Pham-Gia *et al.* (2008) first proposed the definition of the L^1 - distance between more than two pdfs. From this definition, Tai and Pham-Gia (2010) have proposed the concept the cluster width used as a criterion to build CPD. Hierarchical and non-hierarchical approaches based on this criterion have obtained the results that are suitable for certain problems. The algorithm of Chen and Hung (2015) could determine the suitable number of clusters automatically. Many examples show that this algorithm has more advantages than previously proposed methods. However, it is only suitable in case where the overlapping-degree of the pdfs is not too great. Many applications show that the number of clusters determined by this method is incorrect. Based on WCD, the article proposes a fuzzy cluster analysis of pdfs and determines the suitable number of clusters in non-hierarchical approach. All computational problems of proposed algorithms and expressions are tested by Matlab programs. These programs offer effective support for the complicated calculations of the numerical examples.

The remaining part of the paper is arranged as follows. In Section 2, the definition WCD and related concepts is presented. The WCD of two and more than two pdfs is also determined. Section 3 proposes a fuzzy clustering algorithm for pdfs and an algorithm to determine the suitable number of clusters. Section 4 presents the numerical examples in both synthetic data and real data. The final section is destined for the conclusion of the paper.

2 THE WIDTH CLUSTER OF PROBABILITY DENSITY FUNCTIONS

2.1 Some definitions

Let $f_1, f_2, \ldots, f_k$ be pdfs on R^n, $(k \geq 2, n \geq 1)$, and let $F = \{f_1, f_2, \ldots, f_k\}$, $f_{\max}(x) = \max\{F(x)\}$. We have the following definitions.

Definition 1. WCD of F is defined by following expression:

$$w(F) = \int_{R^n} f_{\max}(x)dx - 1. \quad (1)$$

If F has only one pdf then WCD equals 1.

Definition 2. Let $g, g_1, g_2, \ldots, g_{k1}$, and $f_1, f_2, \ldots, f_{k2}$ be the pdfs. The WCD of $\{(g), (f_1, f_2, \ldots, f_{k1})\}$ and $\{(f_1, f_2, \ldots, f_{k1}), (g_1, g_2, \ldots, g_{k2})\}$ are defined by and $w\left[\{f_1, f_2, \ldots f_{k1}\} \cup \{g_1, g_2, \ldots g_{k2}\}\right]$.

From (1), there are some the following results:

i) $w(f_1, f_2, \ldots, f_k)$ is a non-decreasing function in k and $0 \leq w(F) \leq k-1$. The equality on the left occurs when $f_i(x)$, $i = 1, 2, \ldots, k$ is identical and on the right when $f_i(x)$ have disjoint supports. The smaller the cluster width value is, the higher similarity degree of the pdfs.

ii) The relations concerning the WCD of two consecutive clusters that differ from only one element and those of two clusters and their union are obtained as follows:

$w(f_1, f_2, \ldots, f_{k+1}) - w(f_1, f_2, \ldots, f_k) = 1 - \int_{R^n} \min\{h_1(x), f_{k+1}(x)\}dx.$

$w(f_1, f_2, \ldots, f_k) = w(f_1, f_2, \ldots, f_n) + w(f_{n+1}, f_{n+2}, \ldots, f_k) + 1 - B,$

where

$h_1 = \max\{f_1(x), f_2(x), \ldots, f_k(x)\}$, $B = \int_{R^n} \min\{k_1(x), k_2(x)\}dx, n < k,$

$k_1(x) = \max\{f_1(x), f_2(x), \ldots, f_n(x)\}$, $k_2(x) = \max\{f_{n+1}(x), f_{n+2}(x), \ldots, f_k(x)\}$.

iii) For $k = 2$, it is

$$w(f_1, f_2) = \frac{1}{2}\|f_1, f_2\|_1 = \frac{1}{2}\int_{R^n} |f_1(x) - f_2(x)|\, dx. \quad (2)$$

2.2 Determination of WCD

According to (1), WCD is determined by finding the $f_{\max}(x)$ and integrating this function. We consider them in the following two cases:

2.2.1 For one-dimension

In this case, the $f_{\max}(x)$ can be find by the following algorithm:

Step 1. Solve the equations $f_i(x) - f_j(x) = 0, i = 1, 2, \ldots, k-1,\ j = i+1, \ldots, k,$ to find all roots.

Step 2. With root x_{lm} of $f_l(x) - f_m(x) = 0,$ the value $f_l(x_{lm})$ is compared with all the values of $f_j(x_{lm})$, $j \neq l$, m. If it exists $p \neq l$, m such that $f_p(x_{lm}) > f_l(x_{lm})$ then x_{lm} is deleted and x_{lm} is kept for otherwise. Arrange the kept roots in order from small to large, then we have a roots set $B = \{x_1, x_2, ..., x_h\}$.

Step 3. Give i = 1, 2, …, k; j = 1, 2, …, h, the function $f_{\max}(x)$ is determined by the following principles:

If $\max\{f_1(x_1-\varepsilon_1), f_2(x_1-\varepsilon_1), ..., f_k(x_1-\varepsilon_1)\} = f_i(x_1-\varepsilon_1)$

then $f_{\max}(x) = f_i(x)$ for $x \in (-\infty, x_1)$.

If $\max\{f_1(x_j+\varepsilon_2), f_2(x_j+\varepsilon_2), ..., f_k(x_j+\varepsilon_2)\} = f_i(x_j+\varepsilon_2)$, j = 2, 3,…, h–1

then $f_{\max}(x) = f_i(x)$ if $x \in (x_i, x_{i+1})$.

If $\max\{f_1(x_h-\varepsilon_3), f_2(x_h-\varepsilon_3), ..., f_k(x_h-\varepsilon_3)\} = f_i(x_h-\varepsilon_3)$

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 such that:

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

From this algorithm, the Matlab procedure is established to find the $f_{\max}(x)$. When $f_{\max}(x)$ is determined, the WCD is easily calculated by the formula (1), as well as to classify a new element by principle (1).

2.2.2 For multi-dimension

In multi-dimension cases, it should be very complicated to obtain closed expression for $f_{\max}(x)$. The difficulty comes from the various forms of the intersection space curves between the surfaces of pdfs. This problem has been interested by Pham-Gia *et al*. (2006, 2008), Tai and Pham-Gia (2010), Tai (2016). Pham-Gia *et al*. (2016) have been attempted to find $f_{\max}(x)$ function. However, it is only established for some cases of bivariate normal distribution. Here, we do not find expression of $f_{\max}(x)$. WCD is computed 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 used in Section 4.

3 FUZZY CLUSTERING OF PROBABILITY DENSITY FUNCTIONS ON WCD CRITERION

3.1 Some concepts

The separation of k pdfs into c different clusters can be represented by the partition matrix $U = [\mu_{ij}]_{c\times k}$, where $\mu_{ij} \in [0,1]$ is considered as the probability when the jth element is merged into the ith cluster In non-fuzzy clustering, $\mu_{ij} = 1$ when the j th element belongs to the ith cluster and $\mu_{ij} = 0$ in the opposite case.

The representative pdf of $F = \{f_1, f_2, …, f_k\}$ is defined by

$$f_F = \frac{\sum_{j=1}^{k} (\mu_{Fj})^m f_j}{\sum_{j=1}^{k} (\mu_{Fj})^m}. \tag{3}$$

The weighted exponent m of (3) has an effect on the fuzzy degree of result. For m = 1, fuzzy clustering becomes non-fuzzy clustering. In this article, m = 2 in the numerical examples and applications is chosen.

$f_F \geq 0$ for all x and $\int_{R^n} f_F dx = 1$ is proved. Therefore, the representative pdf of a cluster is also a pdf.

3.2 The proposed fuzzy clustering algorithm

Problem: There are k populations $N^{(0)} = \{W_1^{(0)}, W_2^{(0)}, ..., W_k^{(0)}\}$ with the given pdfs $\{f_1, f_2, ..., f_k\}$. These pdfs need to partition into c clusters (c is given) so that the probability for each pdf belongs to its true cluster is greater than the probability that it belongs to others.

Algorithm: The fuzzy clustering algorithm (FCA) is presented by three following steps:

Step 1**.** Initialize the partition matrix $U^{(0)}$ randomly. Find the representative pdf of cluster fv_i by the formula (3) and compute the cluster width between each pdf and each fv_i by the formula (2).

Step 2**.** Update the new partition matrix $U^{(1)}$ by the following principle:

$$\mu_{ij}^{(1)} = \frac{1}{\sum_{j=1}^{c} \left(w(fv_i, f_j) / w(fv_j, f_i) \right)^{2/(m-1)}}, \text{if}$$

$w(f_j, fv_i) > 0, i = 1, 2, ..., c$, and

$\mu_{ij}^{(1)} = 0$ for otherwise.

Step 3. Compute the value $\left\| U^{(1)} - U^{(0)} \right\| = \max_{i,j} \left(\left| \mu_{ij}^{(1)} - \mu_{ij}^{(0)} \right| \right)$.

Repeat Step 2 and Step 3 n times, until the following condition is satisfied

$$\left\| U^{(n)} - U^{(n-1)} \right\| < \varepsilon.$$

At the end of the algorithm, c clusters are received with the probabilities are presented in the final partition matrix.

In the above algorithm, ε is very small number that is chosen arbitrarily. The smaller ε is, the more iterations and computer's time are taken. In numerical example of Section 4, we chose $\varepsilon = 10^{-4}$.

In this algorithm, after an iteration, the specific probability for merging the f_j into cluster c_i is present. When the algorithm ends, we have the final partition matrix $[\mu_{ij}]_{c \times k}$, where each element μ_{ij} is considered as the probability of which the *j*th pdf belongs to the *i*th cluster. Therefore, if $\max_j \{\mu_{ij}\} = \mu_{il}, j = 1, 2, ..., c$, the pdf f_i will be recognized to belong to *l*th cluster.

3.3 Determining a suitable number of clusters

In clustering approaches, one of the most challenging problems is how to determine the suitable number of cluster. In the literature, the number of clusters can be determined by prior knowledge about the data. Tai and Pham-Gia (2010) has proposed the algorithm with a bound on WCD to solve this problem. However, many results for performing with this algorithm are not suitable. Chen and Hung (2015) also gave a method to identify the number of cluster automatically. Although this approach had broken through in problem to find the number of clusters, it still gives non-reasonable result when the overlap of pdfs is larges. Based on WCD, a new approach to determine the suitable number of cluster *c* on WCD is proposed.

Let $F = \left\{ f_1, f_2, ... f_k \right\}$ be the set of *k* pdfs and $Fv^{(t)} = \left\{ fv_1^{(t)}, fv_2^{(t)}, ... fv_k^{(t)} \right\}$ be the sequences on *k* representative pdfs of clusters in the iteration *t*. The algorithm to determine the suitable number of cluster (SNC) is established. This algorithm consists of there following steps:

Algorithm:

Step 1. For $t = 0$, initialize the sequences of representative pdfs of clusters $Fv^{(0)} = \left\{ fv_1^{(0)}, fv_2^{(0)}, ... fv_k^{(0)} \right\} = F = \left\{ f_1, f_2, ... f_k \right\}$.

Step 2. Update the sequences of representative pdfs of cluster by the formula:

$$fv_i^{(t+1)} = \frac{\sum_{j=1}^{k} K_\lambda \left(fv_i^{(t)}, fv_j^{(t)} \right) . fv_j^{(t)}}{\sum_{j=1}^{k} K_\lambda \left(fv_i^{(t)}, fv_j^{(t)} \right)}, i = 1, ..., k, \quad (4)$$

where $\lambda = \frac{w_s}{5}$, $w_s = \frac{1}{\binom{2}{k}} \sum_{i<j} w(fv_i, fv_j)$,

$$K_\lambda(fv_i, fv_j) = \begin{cases} \exp\left(-\frac{w}{\lambda} \right) & \text{if } w(fv_i, fv_j) \le w_s. \\ 0 & \text{if } w(fv_i, fv_j) > w_s. \end{cases}$$

Step 3. Repeat step 2 until $\max_i \{ w(fv_i^{(t)}, fv_i^{(t+1)}) \} < \varepsilon$, with $\varepsilon > 0$ is a very small positive number.

In the above algorithm, after an iteration has finished, each pdf in *F* converges to the representative pdf of the cluster that contains it. When the algorithm stops, the sequences of *c* representative pdfs is obtained. The number of representative pdfs is the suitable number of clusters containing them in the initial iteration of FCA. As a result, the number of clusters and the initial clusters in the first iteration of FCA can be determined.

The programs by Matlab software to perform FCA and SNC algorithms are established. These programs have applied effectively for numerical examples in Section 4. However, data usually contains discrete elements in practice, so we have to estimate the pdfs before clustering. There are many methods to solve this problem in which the kernel function method is the most popular, for example Parzen (1962), Duin (1997), Scott (1992), Martinez and Martinez (2008), and Silverman (2010). In numerical examples, this method is used with the smoothing parameter chosen by the idea of Scott (1992) and the kernel function being the Gaussian.

4 NUMERICAL EXAMPLES

In this section, two numerical examples are present to show the proposed algorithms and to compare it with existing ones K-means (1767), Goh and Vidal (2008), Tai and Pham-Gia (2010), Montanaria and Calò (2013), Chen and Hung (2015). The first example considers 100 uniform pdfs separated off in two groups with 50 pdfs in each. Example 2 applies to images recognition that can interest in many researches in data mining. The results reveal that the clustering performance of our algorithm is better than existing ones.

Example 1. In this example, the synthetic data studied in Goh and Vidal (2008), Montanari and Carlò (2013), Chen and Hung (2015) are reviewed. The data include two classes f_1 and f_2 with 100 uniform pdfs on the interval $[0,1000]$ (see Figure 1). The pdfs of these two classes are defined as follows:

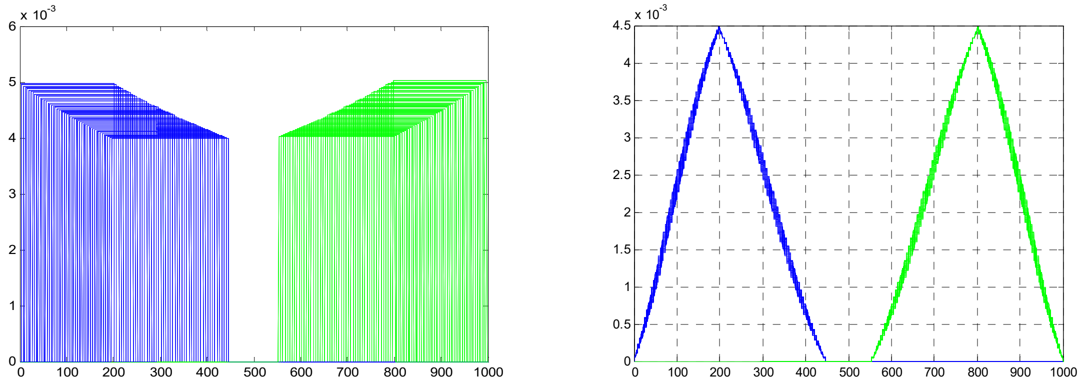
$f_{1,i} = U(a_i, b_i),\ i = 1,\ldots,50$ with

$a_i = 4(i-1) + \lambda_1,\ b_i = 195 + 5i + \lambda_2,$

$f_{2,i} = U(c_i, d_i),\ i = 1,\ldots,50$ with

$c_i = 805 - 5j - \lambda_3,\ d_j = 1004 - 4j - \lambda_4,$

where $U(a_i, b_i)$ and $U(c_i, d_i)$ denote the uniform distribution on the interval (a_i, b_i) and (c_i, d_i), respectively, and $\lambda_1,\ldots,\lambda_4$ are drawn from $U(0,4)$.



(a) (b)

Fig. 1: (a) Graph of the two classes f_1 (left) and f_2 (right)

(b) The final states of f_1 and f_2 after five (convergence) iterations

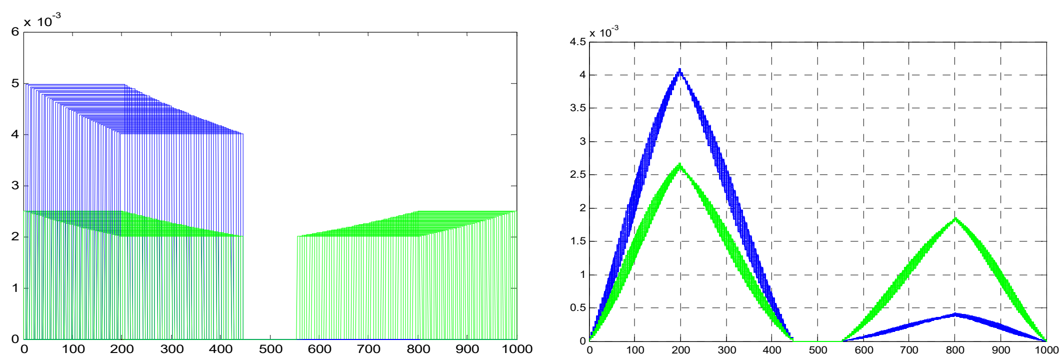
After five iterations, original pdfs converge to two representative pdfs as Figure 1b. Running FCA algorithm after eight (convergence) iterations, the matrix partition (2x100) which its some columns are given below:

$$\begin{bmatrix} 0.768 & 0.768 & 0.795 & \ldots & 0.156 & 0.166 & 0.176 \\ 0.232 & 0.219 & 0.205 & \ldots & 0.844 & 0.834 & 0.824 \end{bmatrix}$$

From the above matrix, we have two clusters $\{f_1, f_2, \ldots, f_{50}\}$ and $\{f_{51}, f_{52}, \ldots, f_{100}\}$. This gives the same result as existing results.

Two classes pdfs g_1 and g_2 ($g_1 = f_1$ and $g_2 = \lambda f_1 + (1-\lambda) f_2, \lambda \in [0, 0.5]$ (Figure 2a) continue to cluster.

After five iterations, the original pdfs converge to two representative pdfs as shown in Figure 2b.



(a) (b)

Fig. 2: (a) The graph of two classes g_1 (left) and g_2 (right) with λ =0.5

(b) The final states of g_1 (left) and g_2 (left) after six (convergence) iterations

FCA algorithm with 24 iterations gives us the partition matrix (2x100), where the probability of the 50 first column of the first row are larger one of the second row and the probability of 50 remainder columns are opposite. The probability assigning elements to two clusters are shown in Figure 3.

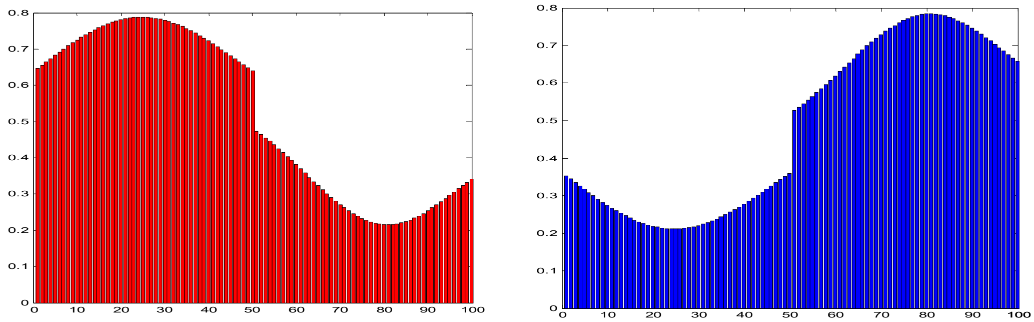


Fig. 3: The graph shows the probabilities of pdfs belong to cluster 1 (lef) and cluster 2 (right)

From partition matrix, there are two clusters: $\{g_1, g_2, ..., g_{50}\}$ and $\{g_{51}, g_{52}, ..., g_{100}\}$. After 500 trials, the error with different values of λ in algorithms of k-Means one, Goh and Vidal (2008), Tai and Pham-Gia (2010), Montanari and Calò (2013), and Chen and Hung (2010) are presented in Table 1.

Table 1: The error (%) of algorithms

Algorithm	$\lambda = 0$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$	$\lambda = 0.4$	$\lambda = 0.5$
k-Means (1967)	49.8	59.8	71.4	78.2	80.6	86.4
Goh and Vidal (2008)	0	0	0	0	0	5.0
Tai and Pham-Gia (2010)	9.2	9.0	9.2	8.8	11.2	13.4
Montanari and Calò (2013)	0	0	0	0	0	5.04
Chen and Hung (2015)	0	0	0	0	0	0
FCA (proposed)	0	0	0	0	0	0

Table 1 gives the best result with Chen and Hung (2015) and the proposed algorithm for all cases of λ with the error 0%.

Example 2. This example considers the problem of image recognition based on its colour 27 images includes 2 categories with 13 images for lotuses and water lily and 14 images for sunflowers are taken. The detail images are given below:

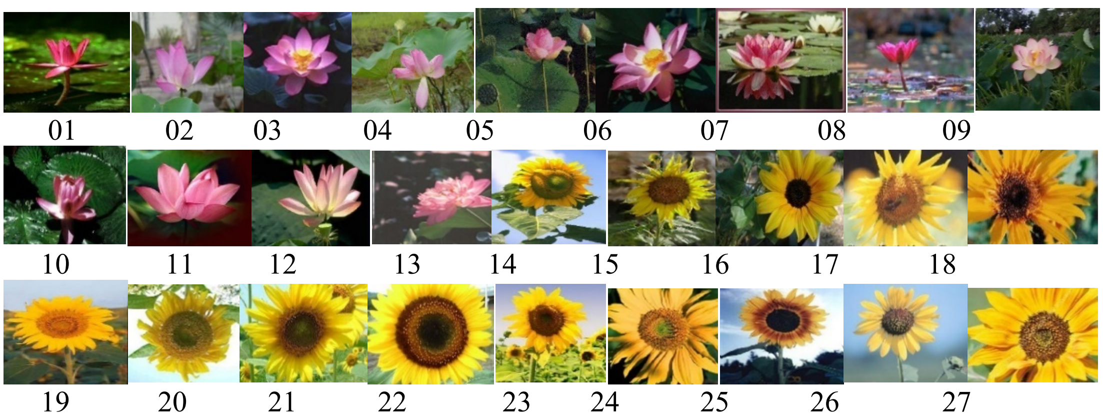


Fig. 4: Images of lotuses, water lily and sunflowers

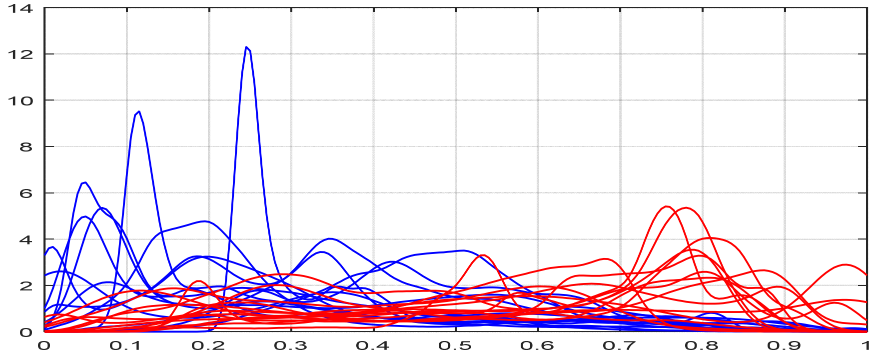
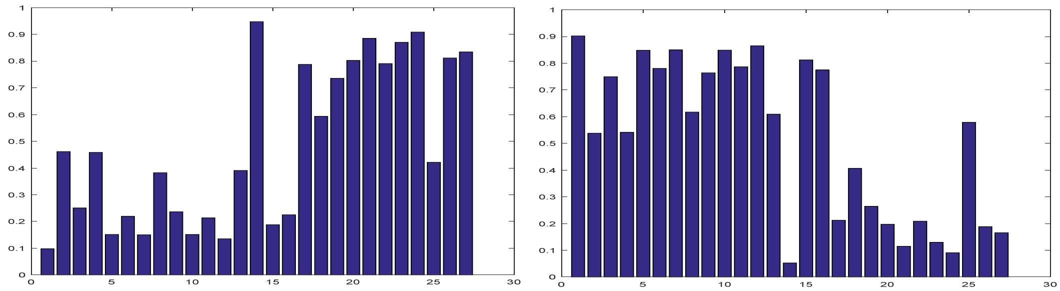


Fig. 5: The estimated pdfs form images of lotuses and sunflowers

Using G scale, the graph of pdfs for images of two classes are given in Figure 5.

SNC algorithm with 6 iterations gives us 2 clusters and after 11 iterations of FCA algorithm, the partition matrix is obtained as follows:

$$\begin{bmatrix} 0.098 & 0.462 & 0.251 & ... & 0.421 & 0.812 & 0.834 \\ 0.902 & 0.538 & 0.749 & ... & 0.579 & 0.188 & 0.166 \end{bmatrix}$$



(a) (b)

Fig. 6: The graph shows the probabilities of pdfs belong to cluster lotuses (a) and cluster sunflowers (b)

From the above matrix, the two clusters are established. Performing 500 trials, the error of result is 11%. Next, cluster analysis is performed for this data set by Chen and Hung algorithm. This algorithm only gives a single cluster with all pdfs after 6 iterations.

ii) Using three variables (R, G, B), making the similar as i), the SNC algorithm gives two clusters after 25 iterations and we obtain the following partition matrix from FCA algorithm:

$$\begin{bmatrix} 0.483 & 0.502 & 0.477 & ... & 0.666 & 0.579 & 0.587 \\ 0.517 & 0.498 & 0.523 & ... & 0.334 & 0.421 & 0.413 \end{bmatrix}$$

With 500 times to perform, we have error 18.5%. Using original data, Chen and Hung (2015) algorithm one time converges one cluster in this case.

From the two above examples, we see that the results of Chen and Hung and the proposed algorithm are the best. However, at the end step, our algorithm gives extra detail the probability which belong to each clusters of elements. We also see that with the example which classes are well separated (Example 1), the result of Chen and Hung (2015) is appreciable. However, in case of overlap region of pdfs are large, this algorithm is disadvantage. Example 2 shows this problem and evidence advantages of the proposed algorithm.

5 CONCLUSIONS

The article has considered the problem to determine the WCD in application for cases of one and multi dimensions. Based on the WCD criterion, a fuzzy cluster algorithm for pdfs and determination of the suitable number of clusters are proposed. These algorithms are applied to several synthetic and real data by Matlab codes. These numerical examples prove suitably and applicability of proposed algorithm. They also show that the results are more efficient than existing ones. In coming, we will apply to other practical issues, especially in data mining with big data. However, the convergence property of proposed algorithms is not studied in this article. Thus, our further studies will emphasize this problem.

REFERENCES

Defays, D., 1997. An efficient algorithm for a complete link method. The computer Journal. 20: 364-366.

Duin, R.P., 1997. On the choice of smoothing parameters for parzen estimators of probability density functions. IEEE Transactions on computers. 25: 1175-1179.

Fukunaga, K., 1990. Introduction to statistical pattern recognition, Second Edition. New York, Academic Press, 616 pages.

Glick, N., 1973. Separation and probability of correct classification among two or more distributions. Annals of the Institute of Statistical Mathematics. 25: 373-382.

Goh, A. and Vidal, R., 2008. Unsupervised Riemannian clustering of probability functions. ECML PKDD, Par 1, LNAI 5211. 2: 377-392.

Chen, H.C. and Hung, L.H., 2015. An automatic clustering algorithm for probability density functions. Journal of statistical computation and simulation. 85: 3047-3063.

Matusita, K., 1967. On the notion of affinity of several distributions and some of its applications. Annals of the Institute of Statistical Mathematics. 19: 181-192.

Martinez, W.L. and Martinez, A.R., 2008. Computational statistics handbook with Matlab. Boca Raton, Chapman & Hall/CRC.

Montanari, A. and Carlò, D.G., 2013. Model-based clustering of probability density functions. Advances in Data Analysis and Classification. 7(3): 301-319.

Parzen, E., 1962. On estimation of a probability density function and mode. Annals of Mathematical statistics. 33: 1065-1076.

Pham-Gia, T., Turkkan, N., and Bekker, A., 2006. Bayesian analysis in the L^1– norm of the mixing proportion using discriminant analysis. Metrika. 64: 1-22.

Pham-Gia, T., Turkkan, N., and Tai, V.V., 2008. The maximum function in statistical discrimination analysis. Commun.in Statistics–Simulation and Computation. 37: 320-336.

Scott, D.W., 1992. Multivariate density estimation: Theory, practice and visualization. John Wiley & Son, New York, 384 pages.

Density estimation for statistics and data analysis. Chapman and Hall. London, 175 pages.

Tai, V.V., and Pham-Gia, T., 2010. Cluster probability distributions. Journal of Applied Statistics. 37: 1891-1910.

Toussaint, G.T., 1971. Some inequalities between distance measures for feature, Evaluation. IEEE Transactions on Computers. 21(4): 409-410.

Webb, A., 2002. Statistical pattern recognition. John Wiley & Sons, New York, 504 pages.