


## fcvalid: An R Package for Internal Validation of Probabilistic and Possibilistic Clustering

 Zeynel Cebeci<sup>1</sup>

<sup>1</sup>Corresponding Author; Div. of Biometry & Genetics, Çukurova University, Adana-Turkey;  
zcebeci@cu.edu.tr; <https://orcid.org/0000-0002-7641-7094>; +903223386084

Received 24 December 2019; Revised 12 April 2020; Accepted 14 April 2020; Published online 30 April 2020

### Abstract

In exploratory data analysis and machine learning, partitioning clustering is a frequently used unsupervised learning technique for finding the meaningful patterns in numeric datasets. Clustering aims to identify and classify the objects or the cases in datasets in practice. The clustering quality or the performance of a clustering algorithm is generally evaluated by using the internal validity indices. In this study, an R package named 'fcvalid' is introduced for validation of fuzzy and possibilistic clustering results. The package implements a broad collection of the internal indices which have been proposed to validate the results of fuzzy clustering algorithms. Additionally, the options to compute the generalized and extended versions of the fuzzy internal indices for validation of the possibilistic clustering are also included in the package.

**Keywords:** internal validity indices, fuzzy clustering, possibilistic clustering, data analysis, R

## fcvalid: Olasılıklı ve Olabilirlikli Bölümleyici Kümelemede Bulanık Geçerlilik İndeksleri için Bir R Paketi

### Öz

Bölümleyici kümeleme, keşifsel veri analizi ve makine öğrenmesinde sayısal veri kümelerindeki anlamlı örüntüleri bulmak için yaygın olarak kullanılan denetimsiz öğrenme tekniklerinden biridir. Kümeleme, pratikte veri kümesindeki nesnelere veya olguları tanımayı ve sınıflandırmayı amaçlar. Bir kümeleme analizinin kalitesi veya bir kümeleme algoritmasının performansı genellikle iç geçerlilik endeksleri kullanılarak değerlendirilir. Bu çalışmada, bulanık ve olabilirlikli kümeleme sonuçlarının doğrulanması için 'fcvalid' adında bir R paketinin işlevleri tanıtılmaktadır. Paket, bulanık kümeleme algoritmalarının sonuçlarını doğrulamak için önerilen çok sayıda iç endeksin uygulamasını içermektedir. Ayrıca, olabilirlikli kümelemenin doğrulanması için bulanık iç endekslerin genelleştirilmiş ve genişletilmiş sürümlerini hesaplama seçenekleri de pakete dâhil edilmiştir.

**Anahtar Kelimeler:** iç geçerlilik endeksleri, bulanık kümeleme, olabilirlikli kümeleme, veri analizi, R

### 1. Introduction

Clustering is one of the frequently used unsupervised learning techniques to explore the meaningful substructures or patterns in examined datasets. The objective of clustering is to divide a dataset into  $c$  subsets by using a clustering algorithm. As a result of clustering, similar set of data points are brought together to form groups or classes so-called clusters. In the related literature, numerous clustering algorithms have been introduced using different approaches to divide a dataset into subsets. These algorithms can primarily be categorized as the hierarchical and the non-hierarchical (or flat) clustering algorithms. The non-hierarchical algorithms can also be further classified into the partitioning algorithms, the density-based algorithms, the grid-based algorithms and the model-based algorithms.

As the subject of this study, the partitioning clustering algorithms assign data points into one of  $c$  clusters, a predefined number of clusters. Then, they iteratively reallocate data points to reach a good quality of clustering result. According to the constraints to define membership degrees of data points to clusters, the partitioning clustering algorithms can be probabilistic, possibilistic and combined version

of probabilistic and possibilistic ones. Further they can be hard and soft in regard of expression of the membership degrees of data points. The well-known K-means algorithm and its successors such as K-medoids PAM, CLARA etc. are the examples of probabilistic partitioning algorithms produce hard clustering results. Here, the term "hard" means that a data point can be a member of only one cluster. But, in reality, some data points can be in an equidistant location to the centres of two or more clusters in a dataset. So they should be member of several clusters with some degrees of membership. The algorithms assigning such fuzzy points to more than one clusters with varying membership degrees are called "fuzzy" or "soft" algorithms.

Fuzzy C-Means Clustering (FCM) algorithm [1] and its modifications which have been developed later are the well-known examples of soft probabilistic partitioning algorithms. However it is sensitive to outliers in datasets. FCM has been the primary algorithm for fuzzy clustering in numerous applications. Krishnapuram and Keller [2-3] developed Possibilistic C-Means (PCM) algorithm. They proposed to relax the probabilistic constraint of FCM in order to fix the outliers problem with FCM. However, if the algorithm poorly initialized, PCM can produce coincident clusters. Later, the mixed algorithms have been proposed by combining FCM and PCM to overcome the issues with FCM and PCM algorithms. The algorithm Fuzzy Possibilistic C-Means (FPCM) [4] was one of the earlier examples of this kind of algorithms. It has been revealed that FPCM algorithm has the row sum constraints problem for the probabilistic part of its objective function. For this reason, Pal et al [5] proposed Possibilistic Fuzzy C-Means (PFCM) to solve the above mentioned problems with FCM, PCM and FPCM. As another algorithm, Possibilistic Clustering Algorithm (PCA) was developed to improve FCM and PCM [6]. Recently, as an improved version of PCA, Wu *et al* [7] introduced Unsupervised Possibilistic Clustering (UPFC) algorithm in order to eliminate the problems such as noise sensitivity and coincident clusters. UPFC has also the advantage that it does not require an FCM initialization for possibilistic part of the clustering.

In partitioning clustering, be either probabilistic or possibilistic, a partitioning task performed with the actual number of clusters in an examined dataset or at least a close value to it, results with a good quality of clustering. Hence, in order to ensure the quality of a clustering analysis, its result should be validated by using the internal fuzzy validity indices. Most of the fuzzy indices have been proposed for validation of the results of the basic FCM algorithm and its successor that can produce fuzzy membership degrees only [8-11]. So, the fuzzy internal indices cannot directly used in validating the possibilistic results. Since various variants of FCM and PCM such as PFCM and UPFC compute both probabilistic membership degrees and possibilistic typicality degrees, the extended and generalized validity indices are needed to simultaneously evaluate the probabilistic and possibilistic clustering results.

As stated by Jain and Dubes [12], the validation of clustering results is the most difficult and deterrent task in cluster analysis. Therefore, while there is a need for development of more effective indices, there is also a strong need for their implementations. Although the availability of some software components and stand-alone tools to be used in fuzzy clustering validation, most of them lack the options that validate possibilistic clustering results. Additionally, most of existing tools only serves a limited number of validity indices for evaluating the result of fuzzy clustering. Therefore, in this study, an R package named 'fcvalid' is introduced as a useful tool to be used for validating the clustering results from FCM, PCM, FPCM, PFCM, UPFC and the other fuzzy and possibilistic clustering algorithms.

## 2. Probabilistic and Possibilistic Partitioning Clustering Algorithms

In this study, FCM, PCM and UPFC were used as the representatives of probabilistic, possibilistic and combined algorithms to test the functions of internal validity indices. In this section, a compact motivation is given to introduce these algorithms. Let  $V = \{v_1, v_2, \dots, v_c\}$  be a prototypes matrix for the cluster centres in dataset  $X = \{x_1, x_2, \dots, x_n\}$  to be partitioned. Here,  $p$  is the number of variables or features,  $c$  is the number of partitions, and  $n$  is the number of data points. Bezdek's original FCM algorithm [1] uses the objective function in Equation 1.

$$J_{FCM}(\mathbf{X}; \mathbf{U}, \mathbf{V}) = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d_{ijA}^2 \quad (1)$$

Although the classical K-means algorithm works with squared distances, the objective function of FCM uses weighted squared distances. In the objective function in Equation 1, a fuzzy partition of  $\mathbf{X}$  is given with the membership matrix  $\mathbf{U}$  of  $n \times c$  dimension.

$$\mathbf{U} = [u_{ij}] \in M_{FCM} \quad (2)$$

In Equation 2,  $u_{ij}$  is the membership degree of  $\mathbf{x}_j$  to the cluster  $i$ . So, the column  $i$  in  $\mathbf{U}$  includes the membership degrees of  $n$  data points to the cluster  $i$ . In Equation 3,  $\mathbf{V}$  is a cluster prototypes matrix.

$$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c], \quad \mathbf{v}_i \in \mathbb{R}^p \quad (3)$$

In Equation 1 above,  $d_{ijA}^2$  is the distance between the center of the cluster  $i$  and the data point  $j$ . As seen in Equation 4, it is calculated as a squared inner-product distance norm.

$$d_{ijA}^2 = \|\mathbf{x}_j - \mathbf{v}_i\|_A^2 = (\mathbf{x}_j - \mathbf{v}_i)^T \mathbf{A} (\mathbf{x}_j - \mathbf{v}_i) \quad (4)$$

In Equation 4, the matrix  $\mathbf{A}$  is symmetric and positive norm matrix. When the matrix  $\mathbf{A}$  equals the unit matrix  $\mathbf{I}$ ,  $d_{ijA}^2$  is computed in squared Euclidean norm. In Equation 1,  $m$  is a weighting exponent which is set to a real number greater than 1. If  $m$  goes to 1 clustering becomes crisper. On the other hand, as it approaches infinity, clustering becomes more fuzzy. The exponent value is generally set to 2 for many applications. The constraints of the objective function of FCM are given in Equation 5.

$$u_{ij} \in [0,1], \forall i, j; \sum_{i=1}^c u_{ij} = 1, \forall j; 0 < \sum_{j=1}^n u_{ij} < n, \forall i \quad (5)$$

FCM is an iterative algorithm whose details are given below.

1. Initialize the matrices  $\mathbf{U}$  and  $\mathbf{V}$ .
2. Update the matrix  $\mathbf{V}$  with Equation 6.

$$\mathbf{v}_i = \frac{\sum_{j=1}^n u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^n u_{ij}^m}; \forall i \quad (6)$$

3. Update the matrix  $\mathbf{U}$  with Equation 7.

$$u_{ij} = \left( \sum_{i=1}^c (d_{ijA}/d_{kjA})^{2/(m-1)} \right)^{-1}; \forall j, k \quad (7)$$

4. If  $\|\mathbf{U}^{(r)} - \mathbf{U}^{(r-1)}\| < \varepsilon$  or  $r > r_{max}$  then stop else go to the step 2.

As seen the algorithm above, FCM updates the matrices  $\mathbf{U}$  and  $\mathbf{V}$  with Equation 6 and Equation 7 at each iteration step. It stops if the number of iterations ( $r$ ) is greater than a user-defined value for maximum number of iterations ( $r_{max}$ ). It also stops when the difference between the sums of objective function in two successive iteration steps is less than a user-defined convergence value ( $\varepsilon$ ).

Possibilistic C-Means (PCM) introduced by Krishnapuram and Keller [2-3] is the first possibilistic algorithm that solves the FCM's problem because of outlier values by omitting the row sum constraint in Equation 5. With PCM algorithm, the data points closer to the cluster centers are evaluated to be "typical" members whereas the data points away from the cluster centers are considered as "atypical" members of the clusters in a dataset. The typicality degrees obtained with PCM range from 0 to 1. A data point having zero and near zero typicality degree is a typical member of a cluster, while those close to one can be considered noise. The objective function of PCM is formulated as given in Equation 8.

$$J_{PCM}(X; T, V) = \sum_{j=1}^n \sum_{i=1}^c t_{ij}^m d^2(\mathbf{x}_j, \mathbf{v}_i) + \sum_{i=1}^c \eta_i \sum_{j=1}^n (1 - t_{ij})^m \quad (8)$$

The row sum constraint of FCM in Equation 5 is not taken into account as the constraint of PCM objective function as seen in Equation 9.

$$\eta_i > 0; \forall i, \quad t_{ij} \in [0, 1]; \forall i, j \quad (9)$$

In Equation 8,  $t_{ij}$  is the typicality degree of  $\mathbf{x}_j$  to the cluster  $i$ . For a good start of PCM, Krishnapuram and Keller [3] suggested to use a clustering configuration obtained from an earlier FCM run. In Equation 8,  $\eta_i$  is a penalty term trying to make  $t_{ij}$  close to 1. It is specifically calculated for each of the clusters in datasets as seen in Equation 10.

$$\eta_i = K \sum_{j=1}^n t_{ij}^m d^2(\mathbf{x}_j, \mathbf{v}_i) / \sum_{j=1}^n t_{ij}^m; K > 0 \quad (10)$$

In Equation 10,  $K$  is a positive number which is generally defined as 1. If  $\eta_i$  is obtained as 0 with Equation 10, PCM gives the same partitioning result with FCM. The updating equations in PCM algorithm are given in Equations 11 and 12.

$$t_{ik} = \left( 1 + \left[ \frac{d^2(\mathbf{x}_j, \mathbf{v}_i)}{\eta_i} \right]^{1/(m-1)} \right)^{-1}; \forall i, j \quad (11)$$

$$\mathbf{v}_i = \frac{\sum_{j=1}^n t_{ij}^m \mathbf{x}_j}{\sum_{j=1}^n t_{ij}^m}; \forall i, j \quad (12)$$

When compared to FCM, for a dataset containing the outliers, PCM is considered more efficient in calculation of the cluster centers during partitioning. But, unfortunately, since it is sensitive to the initial values input for starting the prototype matrix  $\mathbf{V}$ . When the initial values in this matrix are started close to each other, the overlapping clusters may be obtained from a PCM run. As the objective function in Equation 8 approaches a local minimum only some of the centres will be overlapped but PCM has also the other problems. Yang and Wu [6] proposed a possibilistic clustering algorithm named Possibilistic Clustering Algorithm (PCA) which has also the coinciding clusters defect like PCM.

Wu *et al* [7] proposed a clustering algorithm named Unsupervised Possibilistic Fuzzy C-Means (UPFC) in order to eliminate the disadvantages of FCM and PCA. Unlike PCM, UPFC needs not to the matrix  $\mathbf{U}$  returned by a previous FCM analysis since it does not use the sample variances of the features as seen in Equation 13. This results with a remarkable decrease the execution time in cluster analysis.

$$J_{UPFC}(\mathbf{X}; \mathbf{U}, \mathbf{V}) = \sum_{j=1}^n \sum_{i=1}^c (a u_{ij,FCM}^m + b u_{ij,PCA}^\eta) d^2(\mathbf{x}_j, \mathbf{v}_i) + \frac{\beta}{n^2 \sqrt{c}} \sum_{j=1}^n \sum_{i=1}^c (u_{ij,PCA}^\eta \log u_{ij,PCA}^\eta - u_{ij,PCA}^\eta) \quad (13)$$

The constraints of the objective function of UPFC are given in Equation 14.

$$\sum_{i=1}^c u_{ij,FCM} = 1; \forall j; 0 \leq u_{ij,FCM} \leq 1; a > 0; b > 0; m > 1; \eta > 1 \quad (14)$$

As a recent representative of mixed c-means algorithms, UPFC computes both membership and typicality degrees simultaneously. In Equation 13,  $u_{ij,FCM}$  and  $u_{ij,PCA}$  are respectively the fuzzy membership and typicality degrees of  $\mathbf{x}_j$  to the cluster  $i$ . The parameters  $m$  and  $\eta$  are respectively the exponents for fuzziness and typicality, which are set to 2 in general. The values  $a$  and  $b$  in Equation 13 are the weighting coefficients, which defines the relative importance of fuzziness and typicality in the objective

functions of UPFC. If  $b$  is zero, the objective functions of UPFC and FCM are equal to each other. Generally, these coefficients are defined equal to 1.

In order to minimize the objective function of UPFC, the membership degrees ( $u_{ij,FCM}$ ) and typicalities ( $u_{ij,PCA}$ ) are updated as in Equation 15 and Equation 16, respectively.

$$u_{ij,FCM} = \left( \sum_{j=1}^c \left( \frac{d(\mathbf{x}_j, \mathbf{v}_i)}{d(\mathbf{x}_j, \mathbf{v}_j)} \right)^{2/(m-1)} \right)^{-1} \quad \forall i, j \quad (15)$$

$$u_{ij,PCA} = \exp \left( \frac{b n \sqrt{c} d^2(\mathbf{x}_j, \mathbf{v}_i)}{\beta} \right) \quad \forall i, j \quad (16)$$

In Equation 16,  $\beta$  is a variance measure, which is computed using the distances between the overall mean and data points as shown in Equation 17.

$$\beta = \frac{1}{n} \sum_{k=1}^n d^2(\mathbf{x}_k, \bar{\mathbf{x}}); \quad \bar{\mathbf{x}} = \frac{1}{n} \sum_{j=1}^n \mathbf{x}_j \quad (17)$$

Through the iterations the cluster centers are updated by using both  $u_{ij,FCM}$  and  $u_{ij,PCA}$  as formulated in Equation 18.

$$\mathbf{v}_i = \frac{\sum_{j=1}^n (a u_{ij,FCM}^m + b u_{ij,PCA}^\eta) \mathbf{x}_j}{\sum_{k=1}^n (a u_{ij,FCM}^m + b u_{ij,PCA}^\eta)}, \quad \forall i \quad (18)$$

### 3. Internal Validity Indices for Fuzzy Clustering

The internal indices are often used to assess the clustering quality because clustering is an unsupervised learning technique. That is, it is used to determine the clustering pattern in a dataset in which the clustering structure is unknown. Therefore, internal validation quantifies the quality of a clustering relying only on information intrinsic to the examined dataset. It means that the internal indices help to determine the quality of a clustering without respect to any external information.

In the literature, various internal validity indices have been proposed for validating clustering analysis. The detailed information of the internal indices is out of the scope of this study but the details about them can be found in the package manual of 'fcvalid' as well as in some thorough surveys [9-11]. Also the logic behind the indices can be found in the original articles, cited in Table 1. The package 'fcvalid' includes the functions of the internal indices, which are listed in chronological order in Table 1.

Table 1 Internal Validity Indices Implemented in the Package 'fcvalid'

Index	Description
PC	Partition Coefficient [13]
PE	Partition Entropy [13]
APD	Average Partition Density [14]
FHV	Fuzzy Hyper Volume [14]
FS	Fukuyama-Sugeno Index [15]
XB	Xie-Beni Index [16]
AWCD	Average Within-Cluster Distance [17]
K	Kwon Index [18]
CS	Compactness / Separation Ratio [19]
MPC	Modified Partition Coefficient [20]
CWB	Composed Within and Between Scattering Index [21]
SC	Separation/Compactness Ratio [22]
CL	Chen-Linkens Index [23]

Table 1 Internal Validity Indices Implemented in the Package 'fvalid' (cont.)

PBMF	Pakhira-Bandyopadhyay-Maulik Index [24]
TSS	Tang, Sun & Sun Index [25]
FSIL	Fuzzy Silhouette Index [26]
MCD	Minimum Centroid Distance [27]
KPBM	Modified Kernel Form of Pakhira-Bandyopadhyay-Maulik Index [28]

The formulae of the internal validity indices implemented in the package 'fvalid' are given in Table 2. As can be seen from the formulae in Table 2, the indices differ how they measure the compactness (within-cluster variability) and the separability (between-clusters distance). Since clustering aims to maximize compactness and separability, the validity indices try to measure the compactness and separation of clusters after a clustering session. Compactness is a measure how the data points in a cluster are interrelated or adherent to each other. Separation reveals how much a cluster is separated or far from each other. So, the low compactness and high degree of separation indicate a good quality of clustering. Secondly, the internal indices differ which type of information they use in their formulae. However, the majority of them use both the matrices  $U$  and  $V$  in addition to the original dataset  $X$ , a few of them use only the matrix  $U$ .

Table 2 Formulae of the Internal Validity Indices Implemented in the Package 'fvalid'

Index	Formula	Op.V.
PC	$I_{PC}(U) = \frac{1}{n} (\sum_{i=1}^c \sum_{j=1}^n u_{ij}^m)$	max
MPC	$I_{MPC}(U) = 1 - \frac{1}{c-1} (1 - I_{PC})$	max
PE	$I_{PE}(U) = \frac{1}{n} (\sum_{i=1}^c \sum_{j=1}^n u_{ij} \log_b(u_{ij}))$	min
CL	$I_{CL}(U) = \frac{1}{n} \sum_{j=1}^n \max_{1 \leq i \leq c} (u_{ij}) - \frac{1}{\sum_{i=1}^{c-1} i} \sum_{i=1}^{c-1} \sum_{l=i+1}^c (\frac{1}{n} \sum_{l=1}^n \min(u_{ij}, u_{lj}))$	max
FS	$I_{FS}(X; V, U) = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m \ x_j - v_i\ ^2 - \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m \left\  v_i - \frac{1}{c} \sum_{k=1}^c v_k \right\ ^2$	min
XB	$I_{XB}(X; V, U) = \frac{\sum_{i=1}^c \sum_{j=1}^n u_{ij}^2 \ x_j - v_i\ ^2}{n \left( \min_{1 \leq i, k \leq c; i \neq k} \{\ v_i - v_k\ ^2\} \right)}$	min
K	$I_K(X; V, U) = \frac{\sum_{i=1}^c \sum_{j=1}^n u_{ij}^2 \ x_j - v_i\ ^2 + \frac{1}{c} \sum_{i=1}^c \left\  v_i - \frac{1}{n} \sum_{l=1}^n x_l \right\ ^2}{\min_{i \neq k} \{\ v_i - v_k\ ^2\}}$	min
TSS	$I_{TSS}(X; V, U) = \frac{\sum_{i=1}^c \sum_{j=1}^n u_{ij}^2 \ x_j - v_i\ ^2 + \frac{1}{c(c-1)} \sum_{i=1}^c \left\  v_i - \frac{1}{n} \sum_{l=1}^n x_l \right\ ^2}{\min_{1 \leq i \leq c; i \neq k} \{\ v_i - v_k\ ^2\} + \frac{1}{c}}$	min
PBMF	$I_{PBMF}(X; V, U) = \left( \frac{1}{c} \frac{\sum_{j=1}^n u_{1j} \ x_j - \bar{v}\ }{\sum_{i=1}^c \sum_{j=1}^n u_{ij} \ x_j - v_i\ } \max_{1 \leq i, k \leq c; i \neq k} \ v_i - v_k\  \right)^p$	max
KPBM	$I_{KPBM}(X; V, U) = \frac{1}{c \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m \ x_j - v_i\ ^2} \max_{i, k=1, \dots, c; i \neq k} \ v_k - v_i\ ^2$	max
FHV	$I_{FHV}(X; V, U) = \left( \sum_{i=1}^c \det \left[ \frac{\sum_{j=1}^n u_{ij}^m (x_j - v_i)^T (x_j - v_i)}{\sum_{j=1}^n u_{ij}^m} \right] \right)^{1/2}$	min
FSIL	$I_{FSIL}(X; V, U) = \frac{\sum_{i=1}^n (u_{ij} - u_{ij'})^\alpha \left( \frac{b_i - a_i}{\max(b_i, a_i)} \right)}{\sum_{i=1}^n (u_{ij} - u_{ij'})^\alpha}$	min
APD	$I_{APD}(X; V, U) = \frac{1}{c} \sum_{i=1}^c \left( \frac{\sum_{x \in X_j} u_{ij}}{v_i} \right)$	max

Table 2 Formulae of the Internal Validity Indices Implemented in the Package 'fcvalid' (cont.)

CS	$I_{CS}(\mathbf{X}; \mathbf{V}, \mathbf{U}) = \frac{\sum_{j=1}^n u_{ij}^m d^2(\mathbf{x}_j, \mathbf{v}_i)}{\sum_{j=1}^n u_{ij} \sum_{i=1}^c \ \mathbf{v}_i - \mathbf{v}_i\ ^2}$	min
CWB	$I_{CWB}(\mathbf{X}; \mathbf{V}, \mathbf{U}) = \alpha \text{Scat}(c) + \text{Dis}(c)$ $\text{Scat}(c) = \frac{\frac{1}{c} \sum_{i=1}^c \ \sum_{j=1}^n u_{ij} \ \mathbf{x}_j - \mathbf{v}_i\ ^2\ }{\sqrt{\mathbf{X}^T \mathbf{X}}}$ $\text{Dis}(c) = \frac{\max(\ \mathbf{v}_i - \mathbf{v}_k\ )}{\min(\ \mathbf{v}_i - \mathbf{v}_k\ )} \sum_{i=1}^c (\sum_{k=1}^c \ \mathbf{v}_i - \mathbf{v}_k\ )^{-1}$	min
SC	$I_{SC}(\mathbf{X}; \mathbf{V}, \mathbf{U}) = \frac{\sum_{i=1}^c \ \mathbf{v}_i - \bar{\mathbf{v}}\ ^2 / c}{\sum_{i=1}^c (\sum_{j=1}^n u_{ij}^m \ \mathbf{x}_j - \mathbf{v}_i\ ^2 / \sum_{j=1}^n u_{ij})} - \frac{\sum_{i=1}^{c-1} \sum_{k=1}^{c-j} (\sum_{j=1}^n (\min(u_{ij}, u_{ik})^2 / n_{i+k}))}{\sum_{j=1}^n (\max(u_{ij}^2) / \sum_{j=1}^n \max(u_{ij}))}$	max
AWCD	$I_{AWCD}(\mathbf{X}; \mathbf{V}, \mathbf{U}) = \frac{1}{n c} \frac{\sum_{i=1}^c \sum_{j=1}^n u_{ij}^m \ \mathbf{x}_j - \mathbf{v}_i\ ^2}{\sum_{j=1}^n u_{ij}^m}$	min
MCD	$I_{MCD}(\mathbf{V}) = \min_{i,k=1,\dots,c; i \neq k} \ \mathbf{v}_k - \mathbf{v}_i\ ^2$	max

In the formulae in Table 2:

- $\mathbf{v}_i$  : prototype (centres) vector for cluster  $i$ ,
- $\mathbf{x}_j$  : feature vector for data point  $j$ ,
- $d^2(\mathbf{x}_j, \mathbf{v}_i)$  : Euclidean distances between prototype  $\mathbf{v}_i$  and the data point  $\mathbf{x}_j$ ,
- $u_{ij}$  : fuzzy membership degree of data point  $j$  to the cluster  $i$ ,
- $m$  : weighing exponent for fuzziness,
- $c$  : an integer defining the number of clusters to be used in clustering.

A possibilistic algorithm, i.e. PCM, produces only typicality degrees but not membership degrees. Since PCM is free for the row sum constraint in Equation 5, the fuzzy indices in Table 2 become completely useless and do not work properly for typicality degrees. In a pioneer study to validate possibilistic results, Yang and Wu [6] proposed an approach based on normalization of typicality used with the existing fuzzy validity indices.

$$u'_{ij} = \frac{t_{ij}}{\sum_{i=1}^c t_{ij}} ; \forall i, j \quad (19)$$

This technique so-called the generalized index makes the typicality degrees suitable for processing with all of the fuzzy internal indices, as demonstrated for the validity index PE in Equation 20.

$$GI_{PE}(\mathbf{U}') = \frac{\sum_{i=1}^c \sum_{j=1}^n u'_{ij} \log_b(u'_{ij})}{n} \quad (20)$$

The use of normalized  $T$  values with existing internal indices is an option to validate possibilistic clustering results. But other solutions are needed for the results from the mixed c-means algorithms, such as PFCM and UPFC that generate both membership and typicality degrees simultaneously. A limited number of solutions have been proposed for validation of the results from mixed c-means algorithms. For having an idea, here, an example is given for the extended use of XB index in Equation 21. As exemplified for the index XB in Equation 21, the fuzzy indices are extended by using the element-wise sum of  $\mathbf{U}$  and  $\mathbf{T}$  instead using the matrix  $\mathbf{U}$  only. The extended versions of the fuzzy validity indices have already been introduced in detail in a comparative study by Cebeci *et al* [29].

$$EI_{XB}(\mathbf{X}, \mathbf{U}, \mathbf{T}, \mathbf{V}) = \frac{\sum_{i=1}^c \sum_{j=1}^n (u_{ij}^m + t_{ij}^n) \|\mathbf{x}_j - \mathbf{v}_i\|^2}{n \left( \min_{i \neq k} \{\|\mathbf{v}_k - \mathbf{v}_i\|^2\} \right)} \quad (21)$$

#### 4. Demonstration of the Functionality of the Package 'fcvalid'

The package 'fcvalid' includes the functions which are the implementations of a broad collection of the internal indices which are formulated in Table 2 in the previous section. The names of the functions in the package are given in the first column of Table 1. Additionally the package contains two functions named as `allindexes` for listing the values of all of the indices plus the function named `ws` for weighted summation index, which is an ensemble index combining the other indices with certain weights. As usual for every R package, the package 'fcvalid' has also a package manual and vignette describing its functions with the examples.

The recent version of the package 'fcvalid' is distributed on Github repository. In order to install the package from GitHub, at first the 'devtools' package [30] from CRAN should be installed in the local system. Then the package 'fcvalid' is installed by using `install_github` of `devtools` package as shown in the following code chunk in R environment [31].

```
> if(!require(devtools)) {install.packages('devtools'); library(devtools)}  
> install_github('zcebeci/fcvalid')
```

In order to get a compiled version of the vignettes of the package, the package 'fcvalid' alternatively is installed by running `install_github` with `build_vignettes` argument set to `TRUE`. For rendering of the vignette during installation, the package 'rmarkdown' [31] should also be already installed into the local system from CRAN as follows:

```
> install.packages('rmarkdown')
```

After installation of the package 'fcvalid', it is loaded into R working space using `library` or `require` as seen below.

```
> library(fcvalid)
```

In this study, some functionalities of the package is demonstrated with the validation for fuzzy clustering on the iris dataset [33], a well-known real dataset consisting of four features ('Sepal.Length', 'Sepal.Width', 'Petal.Length' and 'Petal.Width') plus a class variable named 'Species' shows the natural classes of three Iris species in the last column. This four-dimensional dataset contains totally 150 data objects, 50 in each class. After loading the dataset into R working space, its last column is removed for applying partitioning clustering on it.

```
> data(iris)  
> x <- iris[,-5]  
> pairs(x, col=iris[,5])
```

The `pairs` function in the package `stats` of R can be used to display the scatterplots between the pairs of features in dataset. Figure 1 illustrates the natural cluster structure in `iris` dataset. This pairwise-scatterplots may also be helpful to compare the existing pattern with the clustering structures obtained with runs of the partitioning algorithms.



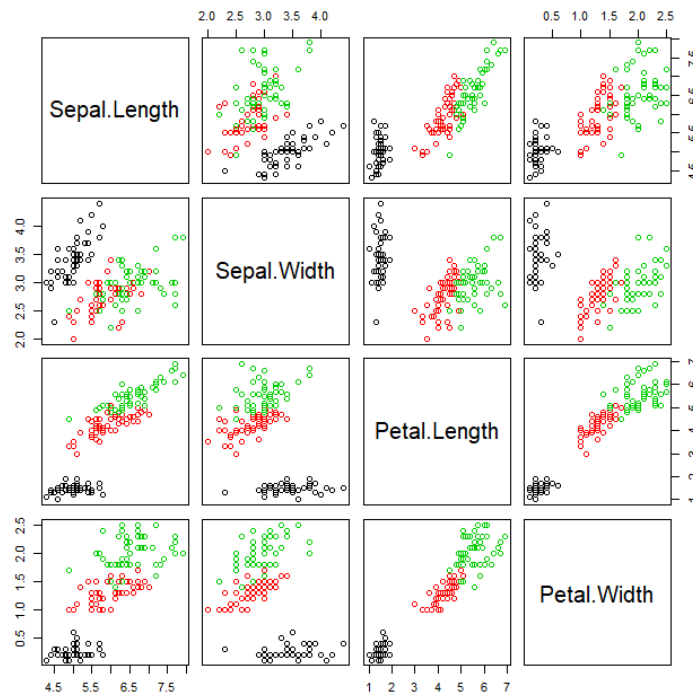


Figure 1 Cluster Structure in Iris Dataset

In R environment, several dozen of R packages are available for almost every kind of clustering methods. In this study, since it has many functions for probabilistic and possibilistic clustering, the R package 'ppclust' [34] are used to get the outputs from the clustering algorithms in order to test the internal indices. If the package 'ppclust' has been already installed in the local system, it can be loaded to R working space with the commands `require` or `library` as follows:

```
> library(ppclust)
```

The functions `fcm`, `pcm` and `upfc` of the package 'ppclust' were used to demonstrate the validation of the results of FCM, PCM and UPFC algorithms, respectively. For this purpose, these functions were called as follows:

```
> resfcm <- fcm(x,centers=3,m=2,nstart=5)
> respcm <- pcm(x,centers=resfcm$v,memberships=resfcm$u, eta=2,nstart=5)
> resupfc <- upfc(x,centers=3,m=2,eta=2,nstart=5)
```

In all of the function calls above, `x` denotes the name of data frame. The fuzziness parameter `m` and the typicality parameter `eta` were set to 2. All of the functions were started 5 times with the `nstart` argument, and the number of clusters `centers` were set to 3. The clustering results obtained from the runs of the functions `fcm`, `pcm` and `upfc` can be displayed with `summary` function of the `ppclust` package. This function can be called for displaying `resfcm`, `respcm` and `resupfc`, which are the cluster objects obtained as the results of the FCM, PCM and UPFC runs in the above examples.

The cluster structure from a clustering analysis can be visually be inspected by using `plotcluster` function of the 'ppclust' package.

```
> plotcluster(resfcm, trans=TRUE)
```



The functions associated with the internal indices in the package 'fcvalid' uses the matrices *u*, *v* and *x* in clustering objects returned by the clustering algorithms. In order to validate the clustering results obtained in the runs of FCM, PCM and UPFC, the related function of a cluster validity indices can be called individually as shown for the indices XB and Kwon in the following code chunk. In the example *x* is name of dataset, *u* is the matrix of fuzzy membership degrees, *v* is the matrix of final cluster centres, *m* is the fuzziness amount, and finally *tidx* is the type of internal index. The default value of *tidx* is "f", stands for fuzzy indices.

```
> xb(x=resfcm$x, u=resfcm$u, v=resfcm$v, m=resfcm$m, tidx="f")
      xb
0.1369082
> kwon(x=resfcm$x, u=resfcm$u, v=resfcm$v, m=2, tidx="f")
      kwon
21.95462
```

Although, the function calls exemplified above can be applied with the results from the other clustering packages of R, they are also more practically called for the results from the package *ppclust* as follows:

```
> xb(resfcm)
      xb
0.1369082
> kwon(resfcm)
      kwon
21.95462
```

However, the clustering quality with a couple of interested internal indices can usually be evaluated as demonstrated above, the validity measures of all of the indices might also be obtained altogether. For this purpose, the function *allindexes* of the package 'fcvalid' can be used as follows:

```
> allindexes(resfcm)
$pc
[1] 0.7833975
$mpc
[1] 0.6750962
$pe
[1] 0.3954916
$xb
[1] 0.1369082
$kwon
[1] 21.95462
$tss
[1] 2.502368
$fs
[1] -1732.456
$pbm
[1] 33.09688
$kpbm
[1] 0.04379026
$awcd
[1] 0.5381878
$cl
[1] 0.7374719
$fhv
[1] 0.04722804
$apd
[1] 3275.88
$sc
[1] 3.205439
$si
[1] 0.8091446
```

```
$cwb
[1] 0.1126691
$cs
[1] 30.34248
```

Since the algorithm UPFC produces both the fuzzy and possibilistic partitions of datasets the internal fuzzy validation indices cannot be directly applied to validate clustering results from this algorithm. The generalized and extended versions of the internal validity indices can be used for validation of the possibilistic clustering results [29]. In the following example, the generalized index values of a UPFC run are computed for the indices XB and Kwon.

```
> xb(resupfc, tidx="g")
  xb.g
0.247465
> kwon(resupfc, tidx="g")
  kwon.g
39.11381
```

Alternatively, the extended index values can be computed to validate the fuzzy and possibilistic clustering results. An extended index value is obtained by summation of fuzzy and possibilistic membership degrees (typicalities) for the algorithms producing both types of partitions. In the package `fcvalid`, the extended index value for an index is calculated by setting the index type argument `tidx` to "e". An extended index values is labeled with an ".e" postfix in the validation results. The following code example demonstrates how to obtain the extended values for the indices XB and Kwon.

```
> xb(resupfc, tidx="e")
  xb.e
0.1613088
> kwon(resupfc, tidx="e")
  kwon.e
50.38669
```

Outputs from the functions of 'fcvalid' can be used to compare the performances of several clustering algorithms, as well as to determine the optimal performance when an algorithm is run with different parameters. It can even be used to compare the efficiencies of the internal validity indices in finding a previously known number of clusters for a given dataset. In order to decide to an optimal clustering result or to find an optimal value of number of clusters in datasets, cluster analysis should be repeated for a range of number of clusters. In the code chunk below, FCM algorithm is run for five different levels of  $c$  (range from 2 to 6). The index values obtained by using the matrices  $U$  and  $V$  from the run, which has the smallest objective function value among the three starts of FCM is seen as the output of after the code chunk.

```
> options(scipen=100, digits=3, width=120)
> c1 <- 2
> c2 <- 5
> indnames <- c("PC", "MPC", "PE", "XB", "K", "TSS", "CL", "FS",
+ "PBMF", "FSIL", "FHV", "APD")
> indvals <- matrix(ncol=length(indnames), nrow=c2-c1+1)
> colnames(indvals) <- indnames
> rownames(indvals) <- paste0("c=", c1:c2)
> i <- 1
> for(c in c1:c2){
+ resfcm <- fcm(x=x, centers=c, nstart=3)
+ indvals[i,1] <- pc(resfcm)
+ indvals[i,2] <- mpc(resfcm)
+ indvals[i,3] <- pe(resfcm)
+ indvals[i,4] <- xb(resfcm)
+ indvals[i,5] <- kwon(resfcm)
+ indvals[i,6] <- tss(resfcm)
+ indvals[i,7] <- cl(resfcm)
+ indvals[i,8] <- fs(resfcm)
```

```

+ indvals[i,9] <- pbm(resfcm)
+ indvals[i,10] <- si(resfcm)$sif
+ indvals[i,11] <- fhv(resfcm)
+ indvals[i,12] <- apd(resfcm)
+ i <- i+1
+ }
> print(t(indvals))

```

	c=2	c=3	c=4	c=5
PC	<b>0.8922</b>	0.7834	0.7068	0.6658
MPC	<b>0.7844</b>	0.6751	0.6091	0.5822
PE	<b>0.1957</b>	0.3955	0.5611	0.6751
XB	<b>0.0542</b>	0.1369	0.1953	0.2277
K	<b>8.3762</b>	21.9546	31.9776	38.2385
TSS	7.8778	<b>5.4255</b>	24.9135	24.5540
CL	<b>0.8657</b>	0.7375	0.6560	0.6202
FS	<b>-1864.9751</b>	-1732.4557	-1607.7954	-1581.0464
PBMF	50.4147	63.7654	<b>68.9058</b>	51.4233
FSIL	<b>0.8845</b>	0.8091	0.7704	0.7632
FHV	<b>0.0357</b>	0.0472	0.0657	0.0906
APD	<b>5220.8579</b>	4012.6392	3119.0546	2588.1829

The index values in each row of the matrix above are checked to find the optimal number of clusters. The number of cluster for the value fits to the lower or upper limits for an index, which are listed in the last column of Table 1, is determined as the optimal number of clusters giving the best clustering configuration. These values are marked in bold in the matrix above. For example, since the maximum value is 0.8922 for the index PC the optimal number of clusters is determined as 2 (column c=2 in the matrix) for this index. According to the results in the matrix of index values, most of the indices proposes 2 as the optimal number of clusters for Iris dataset. There are three classes in iris dataset. But the class 'setosa' is linearly separable from the other two classes while the classes 'versicolor' and 'virginica' are not. Thus most of the internal indices propose the optimal number of clusters as 2 while a few indices can propose as 3. For this reason, a result of 3 shows the good performance of the examined internal validity index. Although it is not an objective of this study, the results show that only the index TSS was discriminated the overlapped clusters while the index PBMF proposed the number of cluster as 4 that is overestimation of the actual number of clusters in the iris dataset.

The computed index values can be visually inspected by using barplots, line graphs or other kinds of graphics. In order to visual inspection of the increasing or decreasing trend of the index values, the code chunk below plots the line graphs of the computed index values as seen in Figure 3.

```

> par(mfrow=c(4,3), mar=c(2,2,1.5,1.5), cex.main=1.2)
> for(i in 1:length(indnames)){
+   plot(0,0, type = "n",
+     cex.lab=0.8, cex.axis=0.8, cex.main=1.2, cex.sub=0.8,
+     xlim = c(1, nrow(indvals)),
+     ylim = c(min(indvals[,i]),max(indvals[,i])),
+     xaxt='n', xlab="number of clusters", ylab="index value",
+     main=indnames[i], sub=" ")
+   axis(side=1, at=seq(1, nrow(indvals), by=1),
+     labels=paste0("c=",c1:c2), col.axis="black", las=1)
+   lines(indvals[,i], type="b", col="blue", lty=1, lwd=2)
+ }

```

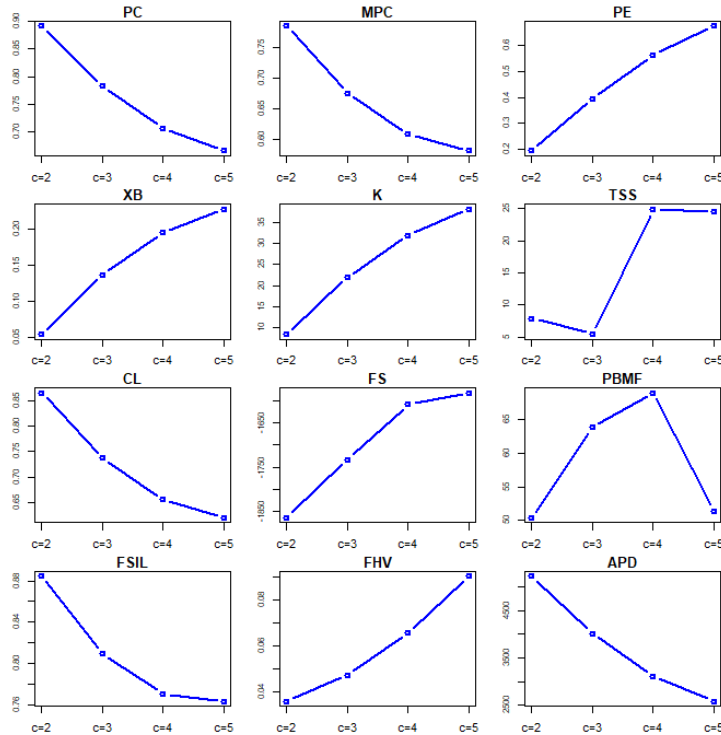


Figure 3 Plots for the Values of Some Internal Indices from the FCM Runs on the Iris Dataset

Barplots are the graphics for easier inspection of the magnitudes of index values. So the following code chunk can be used to plot the barplots of index values as seen in Figure 4.

```
> par(mfrow=c(4,3), mar=c(2,2,1.5,1.5), cex.main=1.5)
> for(i in 1:length(indnames))
+   barplot(indvals[,i], col="dodgerblue", main=indnames[i])
```

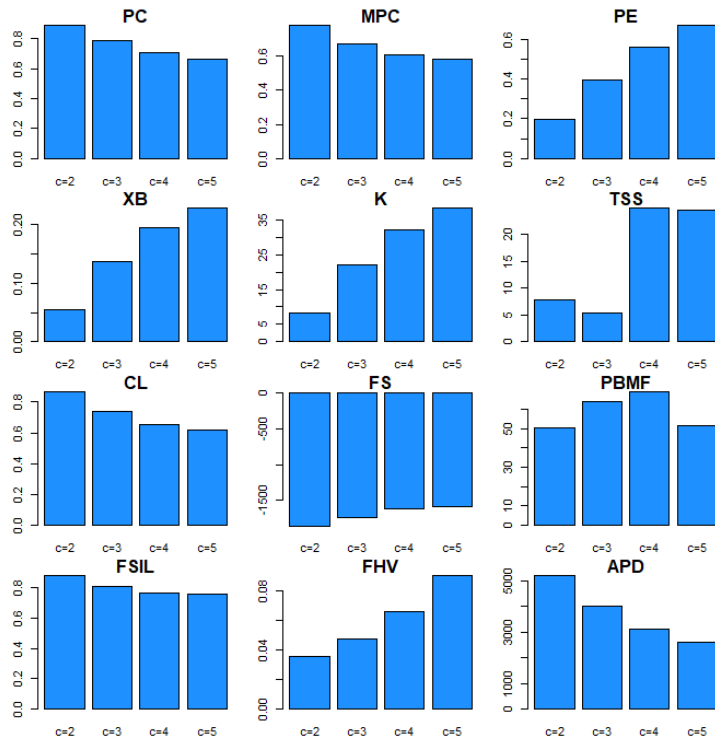


Figure 4 Barplots of the Values of Some Internal Indices from the FCM Runs on the Iris Dataset

## 5. Conclusions

As demonstrated in the previous sections, the R package 'fcvalid' is an all-in-one tool in order to validate the results from the probabilistic and possibilistic clustering algorithms. It provides the implementations of most of the available internal indices for fuzzy clustering validation. Additionally, for evaluation of the possibilistic clustering, the options to compute the generalized and extended versions of the internal indices are also included in the package.

The functionalities of the package makes the researcher to concentrate on substantive issues, without being thinking about the constraints or limitations imposed by the software. Consequently, the package can be used as a test tool for evaluating the performances of partitioning algorithms as well as for finding the optimal number of clusters in fuzzy datasets.

## Acknowledgments

This research was supported by the grant FBA-2019-10285 from the Unit of Scientific Research Projects of Çukurova University, Adana - Turkey.

Supplementary materials including the manual and codes of the package 'fcvalid' can be downloaded from GitHub at <https://github.com/zcebeci/fcvalid>.

## References

- [1] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.
- [2] R. Krishnapuram, J. Keller, "A possibilistic approach to clustering", *IEEE Transactions on Fuzzy Systems*, vol. 1, pp. 98-110, 1993.
- [3] R. Krishnapuram, J. Keller, "The possibilistic c-means algorithm: Insights and recommendations", *IEEE Transactions on Fuzzy Systems*, vol. 4, pp. 385-393, 1996.
- [4] N.R. Pal, K. Pal, J.C. Bezdek, "A mixed c-means clustering model", *Proc. of the 6th IEEE Int. Conf. on Fuzzy Systems*, vol. 1, pp. 11-21, 1997.
- [5] N.R. Pal, K. Pal, J.M. Keller, J.C. Bezdek, "A possibilistic fuzzy c-means clustering algorithm", *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 4, pp. 517-530, 2005.
- [6] K.L. Wu, M.S. Yang, "A cluster validity index for fuzzy clustering", *Pattern Recognition Letters*, vol. 26, no. 9, pp. 1275-1291, 2005.
- [7] X. Wu, B. Wu, J. Sun, H. Fu, "Unsupervised possibilistic fuzzy clustering", *J of Information & Computational Science*, vol. 7, no. 5, pp. 1075-1080, 2010.
- [8] M. R. Rezaee, B. P. Lelieveldt, J. H. Reiber, "A new cluster validity index for the fuzzy c-mean", *Pattern Recognition Letters*, vol. 19, no. 3, pp. 237-246, 1998.
- [9] M. Halkidi, Y. Batistakis, M. Vazirgiannis, "On clustering validation techniques", *J of Intelligent Information Systems*, vol. 17, no. 2-3, pp. 107-145, 2001.
- [10] M. Halkidi, Y. Batistakis, M. Vazirgiannis, "Cluster validity methods: Part I", *ACM Sigmod Record*, vol. 31, no. 2, pp. 40-45, 2002.
- [11] M. Halkidi, Y. Batistakis, M. Vazirgiannis, "Cluster validity methods: Part II", *ACM Sigmod Record*, vol. 31, no. 3, pp. 19-27, 2002.

- [12] A.K. Jain, & R.C. Dubes. *Algorithms for Clustering Data*. Englewood Cliffs: Prentice Hall, 1988.
- [13] J.C. Bezdek, "Cluster validity with fuzzy sets", *J Cybernetics*, vol. 3, no. 3, pp. 58-72, 1974.
- [14] I. Gath, A.B. Geva, "Unsupervised optimal fuzzy clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 773-780, 1989.
- [15] Y. Fukuyama, M. Sugeno, "A new method of choosing the number of cluster for the fuzzy c-means method", *Proc. of the 5th Fuzzy Systems Symp.*, pp. 247-250, 1989.
- [16] X.L. Xie, G. Beni, "A validity measure for fuzzy clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841-847, 1991.
- [17] R. Krishnapuram, C-P. Freg, "Fitting an unknown number of lines and planes to image data through compatible cluster merging", *Pattern Recognition*, vol. 25, pp. 385-400, 1992.
- [18] S.H. Kwon, "Cluster validity index for fuzzy clustering", *Electronics Letters*, vol. 34, no. 22, pp. 2176-2177, 1998.
- [19] A.M. Bensaid, L.O. Hall, J.C. Bezdek, L.P. Clarke, M.L. Silbiger, J.A. Arrington, R.F. Murtagh, "Validity-guided (re)clustering with applications to image segmentation", *IEEE Transactions on Fuzzy Systems*, vol. 4, no.2, pp. 112-123, 1996.
- [20] R.N. Dave, "Validating fuzzy partitions obtained through c-shells clustering", *Pattern Recognition Letters*, vol. 17, pp. 613-623, 1996.
- [21] M.R. Rezaee, B.P. Lelieveldt, J.H. Reiber, "A new cluster validity index for the fuzzy c-mean", *Pattern Recognition Letters*, vol. 19, no. 3, pp. 237-246, 1998.
- [22] N. Zahid, M. Limouri, A. Essaid, "A new cluster-validity for fuzzy clustering", *Pattern Recognition*, vol. 32, no. 7, pp. 1089-1097, 1999.
- [23] M.Y. Chen, D.A. Linkens, "Rule-base self-generation and simplification for data-driven fuzzy models", *Fuzzy Sets and Systems*, vol. 142, no. 2, pp. 243-265, 2004.
- [24] M.K. Pakhira, S. Bandyopadhyay, U. Maulik, "Validity index for crisp and fuzzy clusters", *Pattern Recognition*, vol. 37, no. 3, pp. 487-501, 2004.
- [25] Y. Tang, F. Sun, Z. Sun, "Improved validation index for fuzzy clustering", *Proc. - The American Control Conference, IEEE*, pp. 1120-1125, 2005.
- [26] R.J.G.B. Campello, E.R. Hruschka, "A fuzzy extension of the silhouette width criterion for cluster analysis", *Fuzzy Sets and Systems*, vol. 157, no. 21, pp. 2858-2875, 2006.
- [27] V. Schwaemmle, O.N. Jensen, "A simple and fast method to determine the parameters for fuzzy c-means cluster validation", 2010. [Online]. Available: <http://arxiv.org/abs/1004.1307v1>. [Accessed: 24-Dec-2019].
- [28] A. Chakrabarty, *An Investigation of Clustering Algorithms and Soft Computing Approaches for Pattern Recognition*. PhD. Thesis, Assam Univ., India, 116 p., 2010. [Online]. Available: [http://shodhganga.inflibnet.ac.in/bitstream/10603/93443/16/16\\_chapter%208.pdf](http://shodhganga.inflibnet.ac.in/bitstream/10603/93443/16/16_chapter%208.pdf) [Accessed: 24-Dec-2019].



- [29] Z. Cebeci, A.T. Kavlak, F. Yildiz, “Validation of fuzzy and possibilistic clustering results,” Proc. - *International Artificial Intelligence and Data Processing Symposium IDAP 2017, IEEE*, pp. 1-7, 2017. doi: 10.1109/IDAP.2017.8090183
- [30] H. Wickham, J. Hester, W. Chang, “devtools: Tools to Make Developing R Packages Easier”, R package version 2.2.1, 2019. [Online]. Available: <https://CRAN.R-project.org/package=devtools>, [Accessed: 24-Dec-2019].
- [31] R. Core Team, “R: A language and environment for statistical computing”, *R Foundation for Statistical Computing Vienna Austria*, 2017.
- [32] Y. Xie, J.J. Allaire, G. Golemund, G., *R Markdown: The Definitive Guide*. Chapman and Hall/CRC, 2018. [Online]. Available: <https://bookdown.org/yihui/rmarkdown>, [Accessed: 24-Dec-2019].
- [33] E. Anderson, “The Irises of the Gaspé Peninsula”, *Bull. Amer. Iris Soc.*, vol. 59, pp. 2-5, 1935.
- [34] Z. Cebeci, F. Yildiz, A.T. Kavlak, C. Cebeci, H. Onder, “ppclust: Probabilistic and Possibilistic Cluster Analysis”, R package version 0.1.3, 2019. [Online]. Available: <https://CRAN.R-project.org/package=ppclust>, [Accessed: 24-Dec-2019].