

fclust: an R package for fuzzy clustering

fclust: un pacchetto di R per l'analisi dei gruppi con approccio fuzzy

Maria Brigida Ferraro and Paolo Giordani

Abstract In this work the R package **fclust** is introduced. It contains functions to perform fuzzy clustering analysis. The well-known Fuzzy k -Means algorithm, along with some variants, is available in the package. The package also contains some functions for computing fuzzy cluster validity indices.

Abstract *In questo lavoro viene proposto il pacchetto di R **fclust**. Si compone di funzioni per effettuare una analisi dei gruppi con approccio fuzzy. Il noto algoritmo k -Medie Fuzzy, assieme ad alcune sue varianti, è disponibile nel pacchetto. Il pacchetto contiene anche alcune funzioni per calcolare indici fuzzy per la scelta del numero di gruppi.*

Key words: clustering, fuzzy approach, R, **fclust**

1 Fuzzy clustering algorithms

Let $\mathbf{X}$ be a matrix of order $(n \times p)$ whose generic element x_{ij} , $i = 1, \dots, n$, $j = 1, \dots, p$, is the score of the j -th variable on the i -th object. The Fuzzy k -Means (FkM) algorithm [2] determines the best fuzzy partition of n objects into k clusters by solving the following minimization problem:

Maria Brigida Ferraro

Department of Statistical Sciences, Sapienza University of Rome, P.le A. Moro 5, 00185, Rome
e-mail: mariabrigida.ferraro@uniroma1.it

Paolo Giordani

Department of Statistical Sciences, Sapienza University of Rome, P.le A. Moro 5, 00185, Rome
e-mail: mariabrigida.ferraro@uniroma1.it

$$\begin{aligned}
\min_{\mathbf{U}, \mathbf{H}} J_{\text{FkM}} &= \sum_{i=1}^n \sum_{g=1}^k u_{ig}^m d^2(\mathbf{x}_i, \mathbf{h}_g) \\
\text{s.t. } u_{ig} &\in [0, 1], \sum_{g=1}^k u_{ig} = 1,
\end{aligned} \tag{1}$$

where u_{ig} is the membership degree of object i to cluster g and $\mathbf{h}_g = [h_{g1}, \dots, h_{gp}]$, the g -th row of the prototype matrix $\mathbf{H}$ of order $(k \times p)$, is the prototype of cluster g . The membership degrees u_{ig} , collected in the matrix $\mathbf{U}$ of order $(n \times k)$, indicate the extent to which an object belongs to a cluster. In order to tune the fuzziness of the solution, the fuzziness parameter m is introduced. It takes values strictly higher than 1. When m tends to 1, FkM reduces to the classical Hard k -means.

Although FkM works well in practice, several variants have been proposed in the literature to improve its performance. A recognized limit of FkM is that it is able to produce clusters with spherical shape. In case of non-spherical clusters the Gustafson-Kessel (GK) variant of FkM has been proposed [7]. It replaces the standard Euclidean distance with a cluster-specific Mahalanobis distance:

$$d_V^2(\mathbf{x}_i, \mathbf{h}_g) = (\mathbf{x}_i - \mathbf{h}_g)' \mathbf{V}_g^{-1} (\mathbf{x}_i - \mathbf{h}_g), \tag{2}$$

being $\mathbf{V}_g$ symmetric and positive-definite. It can be interpreted as the covariance matrix for cluster g . If we simply replace $d^2(\mathbf{x}_i, \mathbf{h}_g)$ with (2) in (1) a trivial solution with $\mathbf{V}_g = \mathbf{0}$, $g = 1, \dots, k$ would be obtained since the loss function is linear in $\mathbf{V}_g$. To overcome this problem we must constrain $\mathbf{V}_g$. Gustafson and Kessel [7] suggest to add volume constraints of the form $|\mathbf{V}_g| = \rho_g > 0$ with ρ_g fixed for each g .

The fuzziness parameter m , used in FkM and GK, appears to be an artificial device without a clear physical meaning. In [10, 11] and [12] an entropic variant of FkM is introduced, where the fuzzifying role of m is replaced by adding the Shannon probabilistic entropy as regularization term. It can be interpreted as a measure of entropy for fuzzy sets taking into account that it is applied to the fuzzy membership degrees. The optimization problem is

$$\begin{aligned}
\min_{\mathbf{U}, \mathbf{H}} J_{\text{ent}} &= \sum_{i=1}^n \sum_{g=1}^k u_{ig} d^2(\mathbf{x}_i, \mathbf{h}_g) + p \sum_{i=1}^n \sum_{g=1}^k u_{ig} \log u_{ig} \\
\text{s.t. } u_{ig} &\in [0, 1], \sum_{g=1}^k u_{ig} = 1,
\end{aligned} \tag{3}$$

where p is a non-negative weighting parameter measuring the degree of fuzzy entropy. The degree of fuzzy entropy p is related to the concept of *temperature* in statistical physics [10].

In all of the above algorithms clusters are represented by centroids. The concept of centroids may be not easy to understand for practitioners because centroids are usually artificial objects. For this purpose, it can be helpful to require that the cluster prototypes must be a subset of the observed objects. Such a subset denotes the so-called *medoids*. In this case, the optimization problem can be summarized as:

$$\begin{aligned}
\min_{\mathbf{U}, \bar{\mathbf{X}}} J_{\text{FkMed}} &= \sum_{i=1}^n \sum_{g=1}^k u_{ig}^m d^2(\mathbf{x}_i, \bar{\mathbf{x}}_g) \\
\text{s.t. } u_{ig} &\in [0, 1], \sum_{g=1}^k u_{ig} = 1, \{\bar{\mathbf{x}}_g, l = 1, \dots, k\} \subset \{\mathbf{x}_i, i = 1, \dots, n\}.
\end{aligned} \tag{4}$$

The problem in (4) represents the fuzzy extension of the Hard k -Medoids algorithm [8, 9]. It is well-known that the Fuzzy k -Medoids algorithm is more robust than the standard FkM in presence of outliers.

To deal explicitly with outliers, noise handling variants of FkM have also been proposed. For instance, Davé [4] proposes to consider an additional “cluster”, called noise cluster. If an object is recognized as an outlier, then it is assigned to the noise cluster with a high membership degree. To do it, the following minimization problem must be solved:

$$\begin{aligned}
\min_{\mathbf{U}, \mathbf{H}} J_{\text{Noise}} &= \sum_{i=1}^n \sum_{g=1}^k u_{ig}^m d^2(\mathbf{x}_i, \mathbf{h}_g) + \sum_{i=1}^n \delta^2 \left(1 - \sum_{g=1}^k u_{ig} \right)^m \\
\text{s.t. } u_{ig} &\in [0, 1], \sum_{g=1}^{k+1} u_{ig} = 1.
\end{aligned} \tag{5}$$

In (5) a penalization term depending on a nonnegative parameter δ^2 is added. Such a term is related to the sum of the membership degrees of the objects to the noise cluster, computed as the complement to 1 of the sum of all the membership degrees to the standard k clusters. A partition with $k+1$ clusters is obtained when solving (5). The first k standard clusters are homogeneous, whereas the noise cluster containing the outliers is usually not formed by objects with homogeneous features and, hence, cannot be considered as a cluster in a strict sense.

2 Fuzzy cluster validity indices

In order to choose the number of clusters k a lot of fuzzy cluster validity indices can be used. The most famous ones are probably the partition coefficient (PC) [1] and partition entropy (PE) [2]. They are built on the basis of the membership degree matrix $\mathbf{U}$. Another well-known index is the Xie and Beni (XB) index [13], which measures both compactness and separation of clusters.

Kaufman and Rousseeuw [9] propose the so-called (crisp) Silhouette (CS). For each object i , the average distance of that object to all object belonging to the same assigned cluster g , a_{ig} , and the average distance of object i to all objects belonging to another cluster $f \neq g$, d_{if} , are considered. The silhouette of object i is defined as $s_i = \frac{b_{if} - a_{ig}}{\max(b_{if}, a_{ig})}$, where b_{if} is the minimum distance d_{if} , for all $f = 1, \dots, k, f \neq g$. When cluster contains only a single object i , the silhouette s_i is put equal to 0. In general, $-1 \leq s_i \leq 1$. CS is the mean of the silhouettes s_i 's. The optimal number of clusters k is obtained when CS is maximized. Since CS does not use the member-

ship degree information, Campello and Hruschka [3] propose a fuzzy variant called Fuzzy Silhouette (*FS*). It is defined as

$$FS = \frac{\sum_{i=1}^n (u_{ig} - u_{if})^\alpha s_i}{\sum_{i=1}^n (u_{ig} - u_{if})^\alpha} \quad (6)$$

where u_{ig} and u_{if} are the first and the second largest elements of the i -th column of the membership degree matrix $\mathbf{U}$ and α is a weighting coefficient. As for the crisp version, the optimal number of clusters k is achieved when the index is maximized.

3 The fclust package

The algorithms and cluster validity indices previously recalled are implemented in the R package **fclust** [6]. The standard FkM can be run using the function `FKM`. The function `FKM.ent` performs FkM with entropy regularization and the GK variant is available in the function `FKM.gk` while the function `FKM.gk.ent` performs GK with entropy regularization. The robust algorithms, namely the Fuzzy k -Medoids and the FkM with noise cluster are implemented in the functions `FKM.med` and `FKM.noise`, respectively.

The cluster validity indices can be computed using the functions `PC` (partition coefficient *PC*), `MPC` (a modification of *PC* [5]), `PE` (partition entropy *PE*) and `XB` (Xie & Beni index). The *CS* and *FS* indices are implemented in the functions `SIL` and `SIL.F`, respectively.

The function `Fclust` allows the user to perform an interactive fuzzy clustering analysis. The argument of the function is an object of class `matrix` or `data.frame`. In the following `Fclust` is run in order to apply FKM to the dataset `Mc` (available in **fclust**). The `data.frame Mc` contains data from the McDonald's USA Nutrition Facts for Popular Menu Items:

```
> library(fclust)
> data(Mc)
```

Prior to run `Fclust` we delete the last (qualitative) column and normalize the remaining ones (the nutrition facts) by dividing them by the serving size (first column of `Mc`) which is finally deleted.

```
> for (j in 2:(ncol(Mc)-1))
> Mc[,j]=Mc[,j]/Mc[,1]
> Mc=Mc[,-c(1,ncol(Mc))]
```

We run `Fclust` choosing FkM as clustering algorithm, k equal to 6, m equal to

1.5, RS (number of random starts) equal to 10, and stand equal to 1 (i.e., the algorithm is run using standardized data).

```
> clust=Fclust(Mc)
```

WELCOME to the interactive Fclust program

Warning: If you insert an object of mode CHARACTER when not requested, an error occurs and the program stops!

Specify the clustering algorithm:

```
1: Fuzzy k-means (function FKM)
2: Fuzzy k-means with entropy regularization (function FKM.ent)
3: Gustafson-Kessel-like Fuzzy k-means (function FKM.gk)
4: Gustafson-Kessel-like Fuzzy k-means with entropy regularization
  (function FKM.gk.ent)
5: Fuzzy k-means with noise cluster (function FKM.noise)
6: Fuzzy k-medoids (function FKM.med)
```

```
1: 1
```

Read 1 item

Specify the number of cluster k (default =2):

```
1: 6
```

Read 1 item

Specify the parameter of fuzziness m: (default =2)

```
1: 1.5
```

Read 1 item

Specify the number of random starts RS (default =1):

```
1: 10
```

Read 1 item

Specify the convergence criterion conv (default =1e-9):

```
1:
```

Read 0 items

Specify the maximum number of iterations maxit (default =1e+6):

```
1:
```

Read 0 items

If you want to standardize the dataset before running the clustering algorithm, specify 1:

```
1: 1
```

Read 1 item

The function `Fclust` returns the object `clust` of class `fclust`. It is a list with the following components:

```
> names(clust)
[1] "U" "H" "clus" "value" "cput" "iter" "k" "m" "stand"
   "Xca" "X" "call"
```

`clust$U` and `clust$H` hold the membership degree matrix and the centroid matrix, respectively. `clust$clus` is a matrix containing the indices of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2). `clust$value` holds the loss function values for the RS starts. `clust$cput` and `clust$iter` are the vectors containing the computational times (user times) and the numbers of iterations for the RS starts, respectively. `clust$k` holds the number of clusters. `clust$m` is the parameter of fuzziness. `clust$stand` represents the standardization (Yes if `stand=1`, No if `stand=0`). `clust$Xca` and `clust$X` hold the data used in the clustering algorithm and the raw data, respectively. Finally, `clust$call` is the matched call.

References

1. Bezdek, J.C.: Cluster validity with fuzzy sets. *Journal of Cybernetics* **3**, 58–73 (1974)
2. Bezdek, J.C.: Pattern recognition with fuzzy objective function algorithm, Plenum Press, New York (1981)
3. Campello, R.J.G.B., Hruschka E.R.: A fuzzy extension of the silhouette width criterion for cluster analysis. *Fuzzy Sets and Systems* **157**, 2858–2875 (2006)
4. Davé, R.N.: Characterization and detection of noise in clustering. *Pattern Recognition Letters* **12**, 657–664 (1991)
5. Davé, R.N.: Validating fuzzy partitions obtained through *c*-shells clustering. *Pattern Recognition Letters* **17**, 613–623 (1996)
6. Giordani, P., Ferraro, M.B.: `fclust`: Algorithms for fuzzy clustering and cluster validity indices. R Package version 1.0. (2013), <http://cran.r-project.org/web/packages/fclust/index.html>
7. Gustafson, E., Kessel, W.: Fuzzy clustering with a fuzzy covariance matrix. In: Proc. of IEEE CDC (1978)
8. Kaufman, L., Rousseeuw, P.J.: Clustering by means of medoids. In: Dodge, Y., (ed.) *Statistical Data Analysis Based on the L1 -norm and Related Methods*, pp. 405–416. North Holland, Amsterdam (1987)
9. Kaufman, L., Rousseeuw, P.J.: *Finding Groups in Data: An Introduction to Cluster Analysis*, J. Wiley and Sons (1990)
10. Li, R.P., Mukaidono, M.: A Maximum-Entropy Approach to Fuzzy Clustering. In: *Proceedings of the Fourth IEEE Conference on Fuzzy Systems (FUZZ-IEEE/IFES 95)*, pp. 2227–2232 (1995)
11. Li, R.P., Mukaidono, M.: Gaussian clustering method based on maximum-fuzzy-entropy interpretation. *Fuzzy Sets and Systems* **102**, 253–258 (1999)
12. Miyamoto, S., Mukaidono, M.: Fuzzy *c*-means as a regularization and maximum entropy approach. In: *Proceedings of the Seventh International Fuzzy Systems Association World Congress (IFSA'97)*, II, pp. 8692 (1997)
13. Xie, X.L., Beni, G.: A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**, 841–847 (1991)