An introduction to fuzzy clustering

Maria Brigida Ferraro

Department of Statistical Sciences

SAPIENZA
UNIVERSITÀ DI ROMA

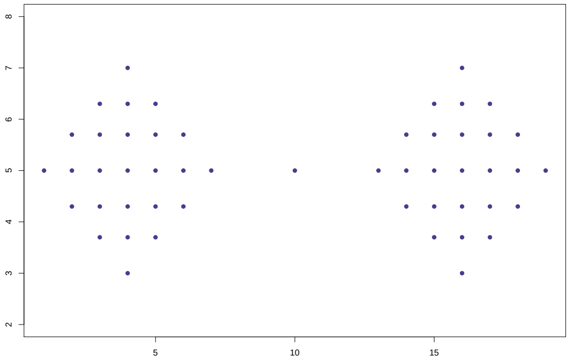
2017 CRoNoS Spring Course on Multivariate methods with R

April 10, 2017 - Cyprus University of Technology

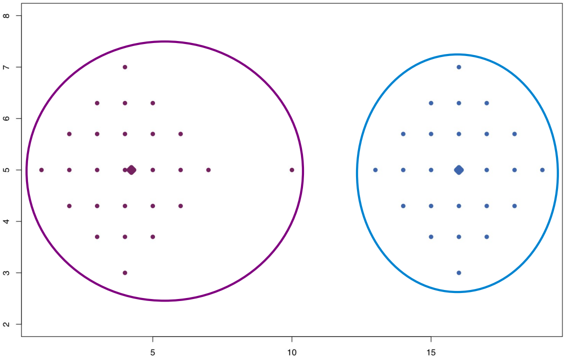
Fuzzy clustering: why?

- The hard (crisp) approach to clustering may fail because its *black-and-white* nature is too rigid to handle the real-life complexity.
- Objects with intermediate characteristics between two clusters are forced to belong to only one cluster.

Motivation example



Motivation example: hard clustering



Fuzzy clustering

✎ Objects are assigned to the clusters according to a certain degree (*grey-scale nature*). It is called membership degree and takes values in [0,1].

Fuzzy clustering: why?

F. Klawonn, R. Kruse, R. Winkler (2015). Fuzzy clustering: more than just fuzzification. *Fuzzy Sets and Systems* 281, 272-279.

- The initial idea of extending the classical k-means clustering technique to an algorithm that uses membership degrees instead of crisp assignments of data objects to clusters led to the invention of a **large variety of new fuzzy clustering algorithms**.

- It has been demonstrated that **the use of membership degrees** for these algorithms - although it is not necessary from the theoretical point of view - **is essential for these algorithms to function in practice**.

Fuzzy k-Means (FkM) (Bezdek, 1974)

$$\min_{\mathbf{U},\mathbf{H}} J_{\mathrm{F}k\mathrm{M}} = \sum_{i=1}^{n}\sum_{g=1}^{k} u_{ig}^{m} d^2\left(\mathbf{x}_i, \mathbf{h}_g\right) = \sum_{i=1}^{n}\sum_{g=1}^{k} u_{ig}^{m} \left\|\mathbf{x}_i - \mathbf{h}_g\right\|^2$$

$$\text{s.t.} \quad u_{ig} \in [0,1], \sum_{g=1}^{k} u_{ig} = 1$$

where

- $\mathbf{X} = [x_{ij}]$: data matrix of order $(n \times t)$
- $\mathbf{U} = [u_{ig}]$: membership degree matrix of order $(n \times k)$
- $\mathbf{H} = [h_{gj}]$: prototype matrix of order $(k \times t)$
- $m(> 1)$: parameter of fuzziness (usually $m = 2$)

with

n: number of objects

t: number of variables

k: number of clusters

FkM: iterative solution (i)

Lagrangian function

$$L = \sum_{i=1}^{n} \sum_{g=1}^{k} u_{ig}^{m} d^{2}\left(\mathbf{x}_i, \mathbf{h}_g\right) - \lambda \left(\sum_{g=1}^{k} u_{ig} - 1 \right)$$

We compute the partial derivatives of L w.r.t. u_{ig} and λ and we set them equal to 0:

$$\frac{\partial L}{\partial u_{ig}} = m u_{ig}^{m-1} d_{ig}^{2} - \lambda = 0$$

$$\frac{\partial L}{\partial \lambda} = \sum_{g=1}^{k} u_{ig} - 1 = 0$$

FkM: iterative solution (ii)

By the usual calculations we then get

$$u_{ig} = \left(\frac{\lambda}{m d_{ig}^2} \right)^{\frac{1}{m-1}}$$

and, taking into account that $\sum_{g=1}^{k} u_{ig} - 1 = 0$,

$$\left(\frac{\lambda}{m} \right)^{\frac{1}{m-1}} = \frac{1}{\sum_{g'=1}^{k} \left(\frac{1}{d_{ig'}^2} \right)^{\frac{1}{m-1}}}$$

FkM: iterative solution (iii)

✎ $u_{ig} = \frac{1}{\sum_{g'=1}^{k} \left(\frac{d^2(\mathbf{x}_i, \mathbf{h}_g)}{d^2(\mathbf{x}_i, \mathbf{h}_{g'})} \right)^{\frac{1}{m-1}}}$

Fixing u_{ig} we obtain $\mathbf{h}_g$ by setting equal to 0 the partial derivatives of L w.r.t. $\mathbf{h}_g$:

$$\frac{\partial L}{\partial \mathbf{h}_g} = 2 \sum_{i=1}^{n} u_{ig}^m d_{ig} = 0$$

✎ $\mathbf{h}_g = \frac{\sum_{i=1}^{n} u_{ig}^m \mathbf{x}_i}{\sum_{i=1}^{n} u_{ig}^m}$

FkM: iterative algorithm

Step 0. Generate randomly a feasible membership degree matrix $\mathbf{U}^{(0)}$.

Step 1. Update the centroid matrix $\mathbf{H}^{(t)}$ keeping fixed $\mathbf{U}^{(t-1)}$.

Step 2. Update the fuzzy membership degree matrix $\mathbf{U}^{(t)}$, keeping fixed $\mathbf{H}^{(t)}$.

Step 3. Check convergence. If the convergence condition is not satisfied, go to *Step 1*.

Convergence criteria

1. Compare the membership degree matrices:

$$\|\mathbf{U}^{(t+1)} - \mathbf{U}^{(t)}\| < \varepsilon$$

   (ε is a fixed value).

2. Compare the centroid matrices

$$\left(\sum_{g=1}^{k} \|\mathbf{h}_g^{(t+1)} - \mathbf{h}_g^{(t)}\|^2 \right)^{\frac{1}{2}} < \varepsilon$$

3. Compare the values of the objective function:

$$J^{(t+1)} - J^{(t)} < \varepsilon$$

But...

- m is a *"strange"* parameter.

- The centroids are computed as weighted means of the data with weights equal to u_{ig}^m rather than u_{ig}.

Entropic Fuzzy k-Means (Li and Mukaidono, 1995, 1999)

$$\min_{\mathbf{U},\mathbf{H}} J_{\mathrm{F}k\mathrm{M.ent}} = \sum_{i=1}^{n}\sum_{g=1}^{k} u_{ig} d^2\left(\mathbf{x}_i, \mathbf{h}_g\right) + p \sum_{i=1}^{n}\sum_{g=1}^{k} u_{ig} \log u_{ig}$$

$$\text{s.t.} \quad u_{ig} \in [0,1], \sum_{g=1}^{k} u_{ig} = 1$$

Iterative solution

$$u_{ig} = \frac{\exp\left(-\frac{d^2(\mathbf{x}_i,\mathbf{h}_g)}{p}\right)}{\sum_{g'=1}^{k}\exp\left(-\frac{d^2\left(\mathbf{x}_i,\mathbf{h}_{g'}\right)}{p}\right)} \qquad \mathbf{h}_g = \frac{\sum_{i=1}^{n} u_{ig}\mathbf{x}_i}{\sum_{i=1}^{n} u_{ig}}$$

✎ p is is the degree of fuzzy entropy and it is called the *“temperature”* in statistical physics.

The EM algorithm for mixture distributions

- A family of mixtures of k density functions contains density of the form

$$p(\mathbf{x}|\theta) = \sum_{g=1}^{k} \alpha_g p_g(\mathbf{x}|\theta_g)$$

- Consider the problem of finding a local minimizer of the function $D(W, \theta)$

$$D(W, \theta) = \sum_{i=1}^{n} \sum_{g=1}^{k} w_{ig} \left(\log w_{ig} - \log \alpha_g p_g(\mathbf{x}_i|\theta_g) \right)$$

EM and Entropic FkM (Hathaway, 1986)

$$D(W,\theta) = E(W) + H(W,\theta)$$

$$E(W) = \sum_{i=1}^{n} \sum_{g=1}^{k} w_{ig} \log w_{ig}$$

$$H(W,\theta) = \sum_{i=1}^{n} \sum_{g=1}^{k} w_{ig} \log \big(1/(\alpha_g p_g(\mathbf{x}_i|\theta_g))\big)$$

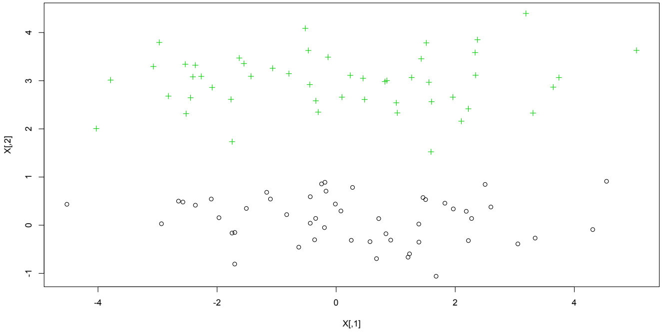
- $E(W)$ is the negative sum of the entropies.
- $H(W,\theta)$ can be viewed as a weighted distances function, with $\log \big(1/(\alpha_g p_g(\mathbf{x}_i|\theta_g))\big)$ representing a probabilistic measure of the distance between $\mathbf{x}_i$ and subpopulation g.

In case of non-spherical clusters?

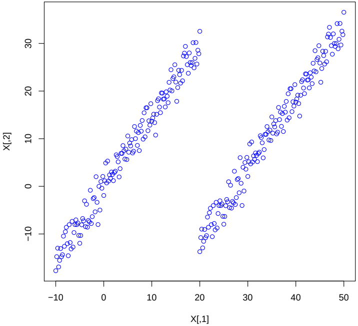
Problem: The FkM and the entropic FkM produce only clusters with spherical shape...

Non-spherical clusters: example 1

$$\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 3 \end{bmatrix} \qquad \Sigma_1 = \Sigma_2 = \begin{bmatrix} 4 & 0 \\ 0 & 0.3 \end{bmatrix}$$



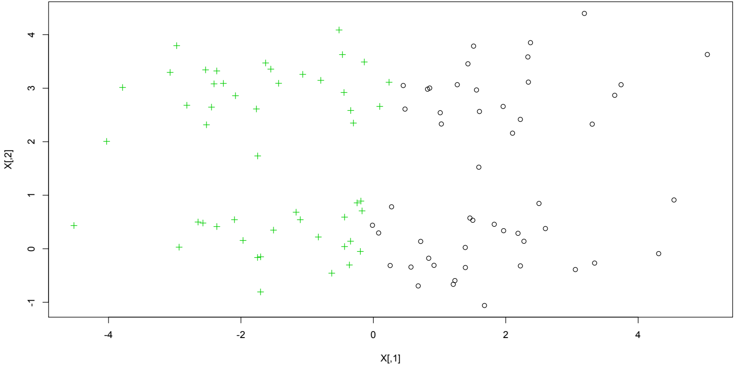
Non-spherical clusters: example 2



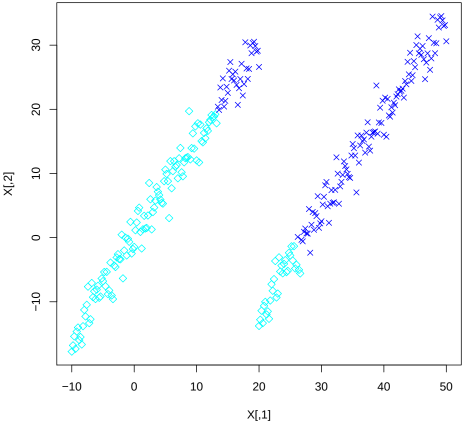
$$\Sigma = \begin{bmatrix} 302 & 146.3 \\ 146.3 & 185.1 \end{bmatrix} \qquad cor(X_1, X_2) = 0.62$$

- non-spherical shapes...high correlations!!!

Example 1: F*k*M - results



Example 2: F*k*M - results



FkM with covariance matrices (Gustafson & Kessel, 1979)

$$\min_{\mathbf{U},\mathbf{H},\mathbf{F}_1\cdots\mathbf{F}_k} J_{\mathrm{F}k\mathrm{M.gk}} = \sum_{i=1}^{n}\sum_{g=1}^{k} u_{ig}^{m} d_M^2\left(\mathbf{x}_i, \mathbf{h}_g\right)$$

$$\text{s.t.} \quad u_{ig} \in [0,1], \sum_{g=1}^{k} u_{ig} = 1, \quad |\mathbf{F}_g| = \rho_g > 0$$

where

- $d_M^2\left(\mathbf{x}_i, \mathbf{h}_g\right) = \left(\mathbf{x}_i - \mathbf{h}_g\right)' \mathbf{F}_g \left(\mathbf{x}_i - \mathbf{h}_g\right)$ is the Mahalanobis distance
- $\mathbf{F}_g$: symmetric and definite positive
- ρ_g: volume parameter (usually equal to 1)

FkM.gk: iterative solution

$$u_{ig} = \frac{1}{\sum\limits_{g'=1}^{k} \left(\frac{d_M^2\left(\mathbf{x}_i, \mathbf{h}_g\right)}{d_M^2\left(\mathbf{x}_i, \mathbf{h}_{g'}\right)} \right)^{\frac{1}{m-1}}} \quad \mathbf{h}_g = \frac{\sum\limits_{i=1}^{n} u_{ig}^m \mathbf{x}_i}{\sum\limits_{i=1}^{n} u_{ig}^m} \quad \mathbf{F}_g^{-1} = \frac{\mathbf{S}_g}{(det(\mathbf{S}_g))^{1/t}}$$

where

$$\mathbf{S}_g = \frac{\sum\limits_{i=1}^{n} u_{ig}^m \left(\mathbf{x}_i - \mathbf{h}_g\right) \left(\mathbf{x}_i - \mathbf{h}_g\right)'}{\sum\limits_{i=1}^{n} u_{ig}^m}$$

is the fuzzy covariance matrix of the g-th cluster

Entropic - FkM - GK
(Ferraro e Giordani, 2013)

$$\min_{\mathbf{U},\mathbf{H},\mathbf{F}_1\cdots\mathbf{F}_k} J_{\mathrm{F}k\mathrm{M.gk.ent}} = \sum_{i=1}^{n}\sum_{g=1}^{k} u_{ig} d_M^2\left(\mathbf{x}_i, \mathbf{h}_g\right) + p\sum_{i=1}^{n}\sum_{g=1}^{k} u_{ig}\log u_{ig}$$

$$\text{s.t.} \quad u_{ig} \in [0,1], \sum_{g=1}^{k} u_{ig} = 1, \quad |\mathbf{F}_g| = \rho_g > 0$$

where

✎ $d_M^2\left(\mathbf{x}_i, \mathbf{h}_g\right) = \left(\mathbf{x}_i - \mathbf{h}_g\right)' \mathbf{F}_g \left(\mathbf{x}_i - \mathbf{h}_g\right)$ is the Mahalanobis distance

FkM.gk.ent: iterative solution

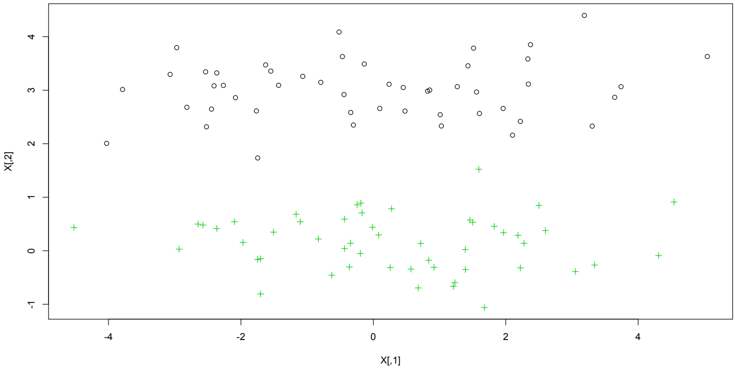
$$u_{ig} = \frac{\exp\left(-\frac{d_M^2(\mathbf{x}_i,\mathbf{h}_g)}{p}\right)}{\sum\limits_{g'=1}^{k} \exp\left(-\frac{d_M^2\left(\mathbf{x}_i,\mathbf{h}_{g'}\right)}{p}\right)} \quad \mathbf{h}_g = \frac{\sum\limits_{i=1}^{n} u_{ig}\mathbf{x}_i}{\sum\limits_{i=1}^{n} u_{ig}} \quad \mathbf{F}_g^{-1} = \frac{\mathbf{S}_g}{(det(\mathbf{S}_g))^{1/t}}$$

where

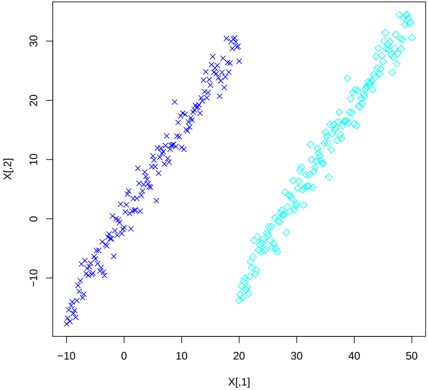
$$\mathbf{S}_g = \frac{\sum\limits_{i=1}^{n} u_{ig}\left(\mathbf{x}_i - \mathbf{h}_g\right)\left(\mathbf{x}_i - \mathbf{h}_g\right)'}{\sum\limits_{i=1}^{n} u_{ig}}$$

is the fuzzy covariance matrix of the g-th cluster

Example 1: F*k*M.gk.ent - results



Example 2: F*k*M.gk.ent - results



Unbalanced clusters

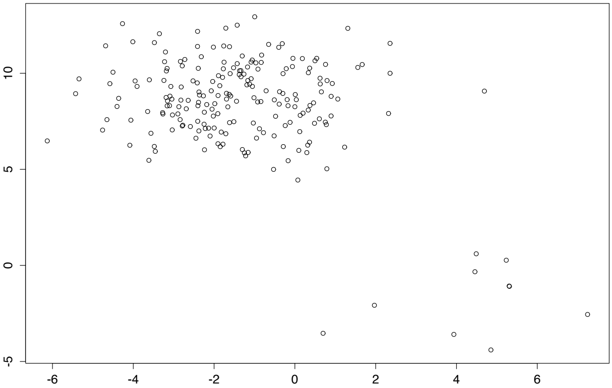


Figure: Example of 2 clusters of size 200 and 10, respectively

F*k*M: results

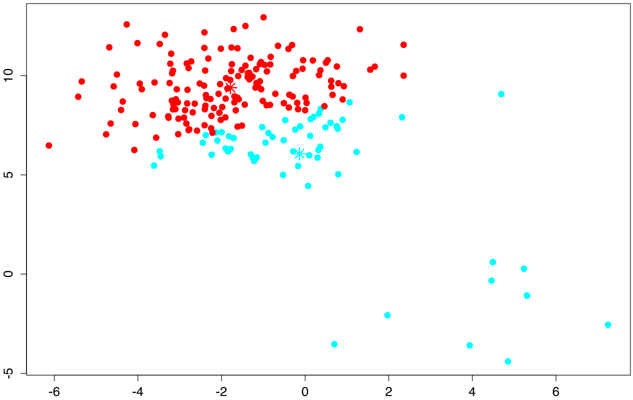


Figure: Example of 2 clusters of size 200 and 10, respectively

Fuzzy clustering with polynomial fuzzifier (Klawonn and Höppner, 2003)

- In F*k*M the **fuzzifier m** is used to control the **overlapping clusters**.
- Furthermore, it leads to assign the **objects to all clusters with non-zero membership degrees**, even when they are very close to the prototypes.

In order to overcome this problem Klawonn and Höppner (2003) propose to use an alternative fuzzifier function: the **polynomial fuzzifier** function.

- In general, a fuzzifier function is a continuous, strictly increasing function $f : [0,1] \longrightarrow [0,1]$ with $f(0) = 0$ and $f(1) = 1$.
- In the F*k*M case, $f(u_{ig}) = u_{ig}^m$.

Fuzzy k-Means with polynomial fuzzifiers (FkM.pf) (Klawonn & Höppner, 2003)

$$\min_{\mathbf{U},\mathbf{H}} J_{\text{F}k\text{M.pf}} = \sum_{i=1}^{n} \sum_{g=1}^{k} f(u_{ig}) d^2 \left(\mathbf{x}_i, \mathbf{h}_g\right)$$

$$\text{s.t.} \quad u_{ig} \in [0,1], \sum_{g=1}^{k} u_{ig} = 1$$

where

- $f(u_{ig}) = \left(\frac{1-\beta}{1+\beta} u_{ig}^2 + \frac{2\beta}{1+\beta} u_{ig}\right)$ is the polynomial fuzzifier fuction
- $\beta \in [0,1]$
  - for $\beta = 0$ we obtain the FkM with parameter m equal to 2
  - for $\beta = 1$ the hard k-means

F*k*M.pf: results

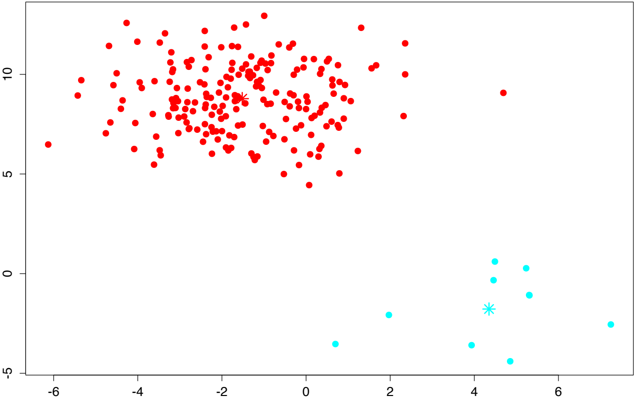


Figure: Example of 2 clusters of size 200 and 10, respectively

FkM.pf: iterative solution

✎ $$u_{ig} = \frac{1}{1-\beta} \frac{1+\beta(k-1)}{\sum_{g'=1}^{k} \frac{d^2\left(\mathbf{x}_i,\mathbf{h}_g\right)}{d^2\left(\mathbf{x}_i,\mathbf{h}_{g'}\right)}} - \frac{\beta}{1-\beta}$$

✎ $$\mathbf{h}_g = \frac{\sum_{i=1}^{n} f(u_{ig})\mathbf{x}_i}{\sum_{i=1}^{n} f(u_{ig})}$$

FkM.pf: algorithm

- The constraint of **non-negative** membership values is **not naturally fulfilled** as it is in FkM.
- To avoid this, in the iterative process we have to take into account the **constraint** $u_{ig} \geq 0$.
- In the membership updates **only the prototypes fulfilling those conditions are included**. The membership for all other prototypes is set to zero. Hence the fuzzy membership value is only computed for a subset of prototypes.

Fuzzy *k*-Medoids (Krishnapuram *et al.*, 2001)

$$\min_{\mathbf{U},\mathbf{M}} J_{\mathrm{F}k\mathrm{M.med}} = \sum_{i=1}^{n} \sum_{g=1}^{k} u_{ig}^{m} d^{2}\left(\mathbf{x}_i, \mathbf{m}_g\right)$$

$$\text{s.t.} \quad u_{ig} \in [0,1], \sum_{g=1}^{k} u_{ig} = 1, \left\{\mathbf{m}_g, g = 1, \ldots, k\right\} \subset \left\{\mathbf{x}_i, i = 1, \ldots, n\right\}.$$

where

- ✎ $\{\mathbf{m}_g, g = 1, \ldots, k\} \subseteq \{\mathbf{x}_i, i = 1, \ldots, n\}$
  the medoids are a subset of the observed objects

The fuzzy *k*-medoids algorithm is usually more **robust** than the standard F*k*M algorithm.

F*k*M.med: iterative algorithm

Step 0a. Initialize the medoid matrix $\mathbf{M}^{(0)}$.

Step 1. Update the fuzzy membership degree matrix $\mathbf{U}^{(t)}$, keeping fixed $\mathbf{M}^{(t-1)}$, by means of

$$u_{ig} = \frac{1}{\sum_{g'=1}^{k} \left(\frac{d^2\left(\mathbf{x}_i, \mathbf{m}_g\right)}{d^2\left(\mathbf{x}_i, \mathbf{m}_{g'}\right)} \right)^{\frac{1}{m-1}}}$$

Step 2. Update the medoid matrix $\mathbf{M}^{(t)}$, keeping fixed $\mathbf{U}^{(t)}$, by using

$$q = argmin_{i'=1}^{n} \sum_{i=1}^{n} u_{ig}^{m} d^2(\mathbf{x}_i, \mathbf{x}_{i'})$$

$g = 1, ..., k$, $\mathbf{m}_g = \mathbf{x}_q$.

Step 3. Check convergence. If the convergence condition is not satisfied, go to *Step 1*.

Initialization

1. Pick all the medoid candidates randomly.

2. Pick the first candidate as the object that is most central to the data set, and then pick each successive one by one in such a way that each one is most dissimilar to all the medoids that have already been picked (see, for more details, Krishnapuram *et al.*, 2001).

3. Pick the first medoid candidate randomly. The rest of the medoids are selected the same way as in Initialization 2.

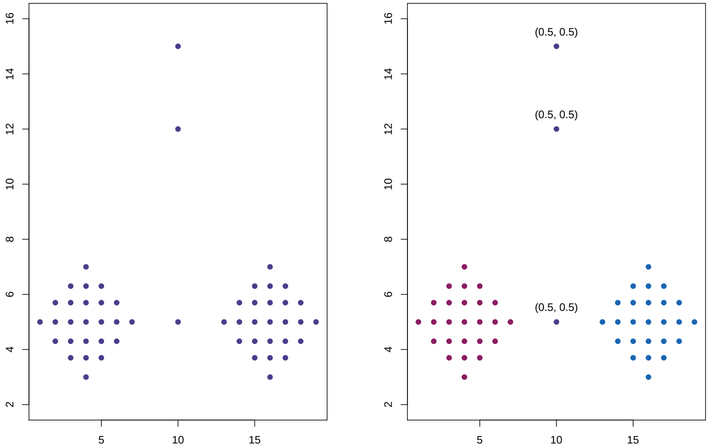
Choice of the fuzziness parameter m

- In the **FkM-type** algorithms the usual choice is m in $[1.5, 2]$.

- Since the medoid has always a membership of one in the cluster, raising its membership to the power m has no effect. Thus, when m is high, the mobility of the medoids from iteration to iteration may be lost. For this reason, we recommend a value between 1 and 1.5 for m for FkM.med.

FkM-type algorithms and Outliers

- The performance of k-means and fuzzy k-means algorithms is **affected by the outliers**.
- The problem is due to the **constraint on the membership degrees**. The sum of the membership degrees of each object to the groups is equal to 1.
- In this way, also the **outliers are assigned to the groups** and the centroids depend on those points.



16
14
12
10
8
6
4
2
5
10
15
(0.5, 0.5)
(0.5, 0.5)
(0.5, 0.5)

"Noise" cluster (Davé, 1991)

In presence of outliers, a possible approach is to consider an additional cluster, called **noise** cluster. If an object is recognized to be an outlier, then it is assigned to the noise cluster with a high membership degree.

"Noise" prototype

A universal entity, $\mathbf{h}_{(k+1)}$, such that it is always at the same distance from every point in the data-set:

$$d(\mathbf{x}_i, \mathbf{h}_{(k+1)}) = \delta, \qquad \forall i.$$

The above definition does not tell us what the distance is. It simply says that **all the points are at the same distance from the noise prototype**.

FkM with noise cluster (Davé, 1991)

$$\min_{\mathbf{U},\mathbf{H}} J_{\mathrm{F}k\mathrm{M.noise}} = \sum_{i=1}^{n}\sum_{g=1}^{k} u_{ig}^{m} d^{2}\left(\mathbf{x}_i,\mathbf{h}_g\right) + \sum_{i=1}^{n} \delta^{2}\left(1-\sum_{g=1}^{k} u_{ig}\right)^{m}$$

$$\text{s.t.} \quad u_{ig} \in [0,1], \sum_{g=1}^{k+1} u_{ig} = 1.$$

where

✎ δ^2: squared distance of each point to the noise cluster

What do we get?

- ✎ A partition with $k+1$ clusters: the first k standard clusters are homogeneous, whereas the noise cluster contains all the outliers and is usually not formed by objects with homogeneous features.

FkM.noise: iterative solution

- $$u_{ig} = \frac{1}{\sum\limits_{g'=1}^{k}\left(\frac{d^2\left(\mathbf{x}_i,\mathbf{h}_g\right)}{d^2\left(\mathbf{x}_i,\mathbf{h}_{g'}\right)}\right)^{\frac{1}{m-1}} + \left(\frac{d^2\left(\mathbf{x}_i,\mathbf{h}_g\right)}{\delta^2}\right)^{\frac{1}{m-1}}}, \quad (g = 1, \cdots, k)$$

- $$u_{i(k+1)} = 1 - \sum_{g=1}^{k} u_{ig}$$

- $$\mathbf{h}_g = \frac{\sum\limits_{i=1}^{n} u_{ig}^m \mathbf{x}_i}{\sum\limits_{i=1}^{n} u_{ig}^m}, \quad (g = 1, \cdots, k)$$

Choice of δ

- If the value of δ is chosen to be very small, then most of the points will get classified as noise points.
- If the value of δ is large, then most of the points will be classified into clusters other than the noise cluster.
- A possible choice is average of the interpoint distances:

$$\delta^2 = \lambda \left(\frac{\sum_{g=1}^{k} \sum_{i=1}^{n} d_{ig}^2}{nk} \right)$$

Cluster validity

- A value representing a measure of the **partition quality** is assigned to the output of a clustering algorithm.
- A **cluster validity** measure is useful to find the optimal values for m and k.

Two types of cluster validity measure:

1. **Fuzziness measure**
2. **Compactness and separation measure**

Fuzziness measure

To evaluate the partition fuzziness, it is necessary to synthesize the information contained in the membership degree matrix in one value. This value indicates the **degree of accuracy of the assignment of units to clusters**.

1. Partition coefficient
2. Partition entropy
3. Modified partition coefficient

Partition coefficient (Bezdek, 1974)

$$PC(k) = \sum_{i=1}^{n} \sum_{g=1}^{k} \frac{(u_{ig})^2}{n}.$$

- The range of variation of PC is $[1/k, 1]$:
  - $1/k$ if all the membership degrees are equal to $1/k$, that is, in case of maximum fuzziness of the partition,
  - 1 if and only if all the membership degrees are equal to 0 or 1 (hard partition).
- The **optimal number** of cluster k is obtained **maximizing** PC w.r.t. k.
- The disadvantages of the partition coefficient are its monotonic tendency and lack of direct connection to some property of the data themselves.

Partition entropy (Bezdek, 1974)

$$PE(k) = -\sum_{i=1}^{n} \sum_{g=1}^{k} \frac{u_{ig} \log_a(u_{ig})}{n}.$$

✎ The range of variation of PE is $[0, \log k]$:

- 0 if and only if all the membership degrees are equal to 0 or 1,
- $\log k$ if all the membership degrees are equal to $1/k$, that is, in case of maximum fuzziness of the partition.

✎ The **optimal number** of cluster k is obtained **minimizing** PE w.r.t. k.

✎ As for PC, there is a monotonic tendency.

Modified partition coefficient (Davé, 1996)

To avoid the monotonic tendency a modified partition coefficient (MPC) has been introduced by Davé. It consists in a linear transformation of PC:

$$MPC(k) = 1 - \frac{k}{k-1}(1 - PC(k)).$$

✎ The range of variation of MPC is $[0, 1]$:

- 0 in case of maximum fuzziness of the partition,
- 1 in case of crisp partition.

Compactness and Separation measures

1. Xie and Beni index
2. Silhouette
3. Fuzzy silhouette

Xie and Beni index (1991)

The Xie and Beni (XB) index is a popular fuzzy cluster validity measure defined as

$$XB(k) = \frac{\sigma_W^2}{d_{min}^2} = \frac{\sum\limits_{i=1}^{n}\sum\limits_{g=1}^{k} u_{ig}^2 d^2\left(\mathbf{x}_i, \mathbf{h}_g\right)}{n \min\limits_{g,g'} d^2(\mathbf{h}_g, \mathbf{h}_{g'})}.$$

- $\sigma_W^2 = \frac{1}{n}\sum\limits_{i=1}^{n}\sum\limits_{g=1}^{k} u_{ig}^2 d^2\left(\mathbf{x}_i, \mathbf{h}_g\right)$ is the total within-cluster distance, that is the **compactness** of the fuzzy partition.
- $d_{min}^2 = \min\limits_{g,g'} d^2(\mathbf{h}_g, \mathbf{h}_{g'})$ is a **separation** measure.

Xie and Beni index: a generalized version

$$XB(k) = \frac{\sigma_W^2}{d_{min}^2} = \frac{\sum_{i=1}^{n} \sum_{g=1}^{k} u_{ig}^m d^2\left(\mathbf{x}_i, \mathbf{h}_g\right)}{n \min_{g,g'} d^2(\mathbf{h}_g, \mathbf{h}_{g'})}.$$

- The compactness is defined as the distance between the objects and the centroids, weighted with u_{ig}^m.
- The separation is defined as the minimum distance between the centroids.
- The **optimal partition** (maximum compactness and separation), is obtained **minimizing** the index.
- The drawback is the monotonic tendency.

Silhouette (Kaufman e Rousseeuw, 1990)

For each object i the silhouette index is defines as:

$$s_i(k) = \frac{b_i - a_i}{\max(b_i, a_i)}.$$

- $a(i)$ is the average distance of object i to all other objects belonging to the same cluster.
- $b(i)$ is the minimum average distance of object i to all objects belonging to another cluster.
- $s_i(k)$ ranges in $[-1, 1]$.

Silhouette index for each object

- If s_i tends to 1 the object i is well assigned to the corresponding cluster. This happens when b_i is high and a_i is small.
- In general, if $a_i < b_i$ then $s_i(k) > 0$. On the other hand, a negative value of $s_i(k)$ indicates that object i is not well assigned to the cluster.

Silhouette index for the partition

The (crisp) silhouette (CS) is defined as the average of s_i over $i = 1, \cdots, n$

$$CS(k) = \frac{1}{n} \sum_{i=1}^{n} s_i(k).$$

The **optimal value** of k is obtained **maximizing** CS.

Fuzzy Silhouette (Campello e Hruschka, 2006)

Since CS does not take into account the membership degree matrix **U**, a fuzzy version of the index has been introduced:

$$FS(k) = \frac{\sum_{i=1}^{n} (u_{ig} - u_{ig'})^{\alpha} s_i(k)}{\sum_{i=1}^{n} (u_{ig} - u_{ig'})^{\alpha}}$$

where

- u_{ig} and $u_{ig'}$ are the first and second largest elements of the i-th row of the fuzzy partition matrix, respectively.
- α is the weight (usually $\alpha = 1$).

The **optimal value** of k is obtained **maximizing** FS.

Visualization techniques (i)

Since the cluster validity indices reduce the information of a large dataset to a **single value**, it is necessary to consider visualization techniques for fuzzy clustering, involving **different information** about the results.

✎ VIFCR (Klawon *et al.*, 2003)

- A chart diagram of the scaled frequency related to the membership degrees

$$\frac{1}{n}\sum_{(i,g):a\leq u_{ig}<b}\left(\frac{k(k-2)}{k-1}u_{ig}+\frac{k}{k-1}\right),$$

with $a, b \in [0,1]$ and $a < b$.

Visualization techniques (ii)

- A diagram whose coordinates, for each object (point) x_i, are
  - u_{ig_1}: the highest membership degree of x_i
  - u_{ig_2}: the second highest membership degree of x_i

  All the points are included in the triangle of vertices $(0,0)$ (noise data), $(0.5,0.5)$ (ambiguous data) and $(1,0)$ (crisp assignments).
- A diagram whose coordinates, for each object (point) x_i, are

$$(d_{ig}, u_{ig})$$

  The ideal situation is to obtain high membership degrees for small distances and low membership degrees for large distances.

Visualization techniques (iii)

✎ VAT (Bezdek & Hataway, 2002)

- The matrix of dissimilarities between the objects, $R = [r_{ij}]$, is considered.
- The matrix is reordered obtaining R^*.
- Its image $I(R^*)$ is displayed.

The number of **dark blocks** along its main diagonal represents the number of **clusters** and the size of each block the approximate size of the cluster.

Visualization techniques (iv)

✎ VCV (Hathaway & Bezdek, 2003)

- First of all the clusters are ordered and the objects in each cluster are ordered by taking into account the membership degrees.
- Then, the dissimilarities r_{ij} between object x_i and x_j are taken into account.
- The following dissimilarities are used:

$$r_{ij}^* = \min_{1 \le g \le k} \{d_{ig} + d_{jg}\},$$

  where $d_{ig} = d(x_i, h_g)$.
- Finally, the information is displayed as an intensity image $I(R^*)$.

Visualization techniques (v)

✎ VCV2 (Huband & Bezdek, 2008)

- The membership degrees matrix U is reordered using the index array of R^* obtained by means of the VAT.
- The resulting matrix $\widehat{U}$ is transformed to the square matrix

$$U^* = \mathbf{1}_n - \left(\widehat{U}^T \widehat{U} / \max\{ \left(\widehat{U}^T \widehat{U} \right)_{ij} \} \right).$$

- The display image $I(U^*)$ is compared with $I(R^*)$ to check the adequacy of the number of clusters.

Software

Starting from F*k*M, fuzzy clustering has received an increasing attention by researchers from several fields.

Nonetheless, popular commercial software solutions (SAS, SPSS, ...) do not contain routines for fuzzy clustering. Just a few exceptions (limited to F*k*M): MATLAB and R.

R package **fclust**, version 1.1.2

Suit of functions for fuzzy clustering analysis (algorithms, cluster validity indices and visualization tools).

▷ http://cran.r-project.org/web/packages/fclust/index.html

M.B. Ferraro, P. Giordani (2015). A toolbox for fuzzy clustering using the R programming language. *Fuzzy Sets and Systems* 279, 1-16.

The package

Package ‘fclust’

September 7, 2015

Type Package

Title Fuzzy Clustering

Version 1.1.2

Date 2015-09-06

Author Paolo Giordani, Maria Brigida Ferraro

Maintainer Paolo Giordani `<paolo.giordani@uniroma1.it>`

Description Algorithms for fuzzy clustering, cluster validity indices and plots for cluster validity and visualizing fuzzy clustering results.

Depends R(>= 2.8.1), base, stats, graphics, grDevices

License GPL (>= 2)

LazyLoad yes

Repository CRAN

NeedsCompilation no

Date/Publication 2015-09-07 00:41:04

fclust

Main features of the package

- 36 functions + 4 datasets
- Most relevant functions for algorithms:

  `FKM`: standard F*k*M algorithm
  `FKM.gk`: Gustafson and Kessel extension of F*k*M
  `FKM.med`: fuzzy *k*-medoids algorithm
  `FKM.noise`: F*k*M with noise cluster
- Most relevant functions for cluster validity indices:

  `PC`: partition coefficient
  `PE`: partition entropy (PE);
  `XB`: Xie and Beni index (XB)
  `SIL.F`: fuzzy silhouette (FS)
- Interactive fuzzy clustering analysis by means of the function `Fclust`

Input arguments (for the algorithms)

- `X`: object of class `matrix` or `data.frame`
- `k`: number of clusters (default: 2)
- `m`: parameter of fuzziness (default: 2)
- `stand`: if `stand=1`, the clustering algorithm is run using standardized data (default: no standardization)
- `RS`: number of (random) starts (default: 1)
- `startU`: rational starting point for the membership degree matrix **U** (default: no rational start)
- `conv`: convergence criterion (default: 1e-9)
- `maxit`: maximum number of iterations (default: 1e+6)
- ...

Output values (for the algorithms)

Object of class `fclust`. List with the following components:

- `U`: membership degree matrix
- `H`: prototype matrix
- `clus`: matrix containing the indices of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
- `medoid`: vector containing the indices of the medoid objects
- `value`: vector containing the loss function values for the `RS` starts
- `cput`: vector containing the computational times (user times) for the `RS` starts
- `Xca`: data used in the clustering algorithm (standardized data if `stand=1`)
- `X`: raw data
- `call`: matched call
- ...

McDonald's data

McDonald's USA Nutrition Facts (81 menu items, no beverages)

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

Variables

- `numeric`:
  `Serving Size`, `Calories`, `Total Fat (g)`, `Saturated Fat (g)`, `Trans Fat (g)`, `Cholesterol (mg)`, `Sodium (mg)`, `Carbohydrates (g)`, `Dietary Fiber (g)`, `Sugars (g)`, `Protein (g)`, `Vitamin A (%DV)`, `Vitamin C (%DV)`, `Calcium (%DV)`, `Iron (%DV)`
- `factor`:
  `Type` (levels: `Burgers & Sandwiches`, `Chicken`, `Breakfast`, `Salads`, `Snacks & Sides`, `Desserts/Shakes`)

Aim of the analysis

Clustering of the menu items (scores normalized w.r.t. `Serving Size`) to discover whether a cluster structure exists (i.e. similar menu items in terms of their nutrition facts) and, in particular, whether a six-cluster structure is visible emerging a link between the variable `type` and the typology of nutrition facts.

Standard F*k*M algorithm (function `FKM`):

```
> fkm   <-   FKM(X = Mc[,1:(ncol(Mc)-1)], k = c,
             m = 1.5, stand = 1, RS = 10)
```

Number of clusters

FS index for values of $k = 2, \ldots, 10$:

`FS` vector containing the FS values (script omitted)

```
> round(FS, 2)
  k = 2  k = 3  k = 4   k = 5  k = 6
   0.52   0.49   0.48    0.55   0.62
  k = 7  k = 8  k = 9  k = 10
   0.64   0.57   0.62    0.61
```

Solution with $k = 7$ clusters (two low-size clusters)

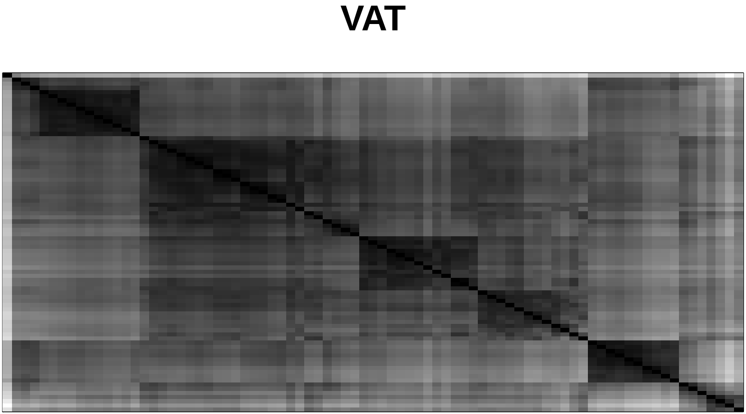
```
> fkm7  <-  FKM(X = Mc[,1:(ncol(Mc)-1)], k = 7,
            m = 1.5, stand = 1, RS = 10)

> cl.size(fkm7$U)
  Cl 1  Cl 2  Cl 3  Cl 4  Cl 5  Cl 6  Cl 7
    24    12     4    13    15    10     3
```

Data Visualization: VAT

Function `VAT(Xca)`

```
> VAT(fkm7$Xca)
```



FkM with $k = 6$ clusters

Trying to avoid low-size clusters, we move to $k = 6$ solution ($FS = 0.62$)

```
> fkm6    <-    FKM(X = Mc[,1:(ncol(Mc)-1)], k = 6,
                m = 1.5, stand = 1, RS = 10)

> cl.size(fkm6$U)
  Cl 1  Cl 2  Cl 3  Cl 4  Cl 5  Cl 6
    12    26    10    15     5    13
```

Comparison between the solutions with $k = 6$ and $k = 7$

```
> table(fkm6$clus[,1], fkm7$clus[,1])

        Cl 1 Cl 2 Cl 3 Cl 4 Cl 5 Cl 6 Cl 7
  Cl 1     0   12    0    0    0    0    0
  Cl 2    24    0    0    0    0    0    2
  Cl 3     0    0    0    0    0   10    0
  Cl 4     0    0    0    0   15    0    0
  Cl 5     0    0    4    0    0    0    1
  Cl 6     0    0    0   13    0    0    0
```

Interpretation of the clusters (i)

```
> table(Mc$Type, fkm6$clus[,1])
```

	Cl 1	Cl 2	Cl 3	Cl 4	Cl 5	Cl 6
Breakfast	12	5	0	1	1	0
Burgers & Sandwiches	0	10	0	0	0	12
Chicken	0	4	0	0	0	0
Desserts/Shakes	0	0	0	12	4	0
Salads	0	0	10	0	0	0
Snacks & Sides	0	7	0	2	0	1

Clusters

Cluster 1 `Breakfast`

(Bacon, Egg & Cheese Biscuit)

Cluster 3 `Salads`

(Premium Southwest Salad with Grilled Chicken)

Interpretation of the clusters (ii)

Cluster 4 — Desserts/Shakes (ice-creams and fruits)

(McFlurry with OREO Cookies)

Cluster 5 — Desserts/Shakes (cookies and pies)

(Oatmeal Raisin Cookie)

More complex interpretation for Clusters 2 and 6

- `Burgers & Sandwiches` assigned to Cluster 6 (although no one-to-one relationship)
- Cluster 2 contains food items of different types

Nonetheless, by further inspecting the food items of type `Burgers & Sandwiches` assigned to Cluster 2 (the code is omitted), a clear interpretation of Clusters 2 and 6 can be found

Interpretation of the clusters (iii)

Findings

- Chicken-made food items belong to Cluster 2 along with two other food items with fish and pork
- All the food items assigned to Cluster 6 contain beef
- 6 (out of 7) food items of type `Snacks & Sides` assigned to Cluster 2 are chicken-based

Hence

Cluster 2 "chicken-made food items"

(Premium Crispy Chicken Ranch)

Cluster 6 "beef-made burgers and sandwiches"

(McDouble)

Centroids (i)

```
> fkm6$Hraw <- Hraw(fkm6$X, fkm6$H)
```

`Breakfast` items have the highest values of `Cholesterol (mg)` and `Sodium (mg)` (a lot of items with eggs)

“chicken-made food items” presents average values for the nutrition facts except for high values of `Sodium (mg)` and the lowest values of `Vitamin A (%DV)`

`Salads` are the most healthy items (the lowest values of `Calories`, `Total Fat (g)`, `Saturated Fat (g)` and `Trans Fat (g)` and the highest values of `Vitamin A` and `Vitamin C (%DV)`)

Centroids (ii)

Ice-creams and fruits (Desserts/Shakes) present the lowest values of Cholesterol (mg), Sodium (mg), Dietary Fiber (g), Protein (g) and Iron (%DV) and the highest values of Calcium (%DV)

Cookies and pies (Desserts/Shakes) are the less dietetic ones: the highest amounts of Calories, Total Fat(g), Saturated Fat (g), Carbohydrates Sugars(g). Also the highest values of Iron (%DV) and the lowest values of Calcium (%DV)

“beef-made burgers and sandwiches” present the highest values of Trans Fat (g) and Protein (g)

Membership degrees (examples)

Oatmeal Raisin Cookie (Cluster 5 with membership degree = 0.99)

Baked Hot Apple Pie (Cluster 5 with membership degree = 0.53)

Mean values (the most relevant variables)

```
> round(apply(fkm6$X[,c(1,2,3,7,9,13,14)],2,mean),2)
 Calories   Total Fat   Saturated Fat   Carbohydrates   Sugars   Iron (%DV)   Calcium(%DV)
     2.33        0.11            0.04            0.25     0.08         0.09           0.08
```

Centroid of Cluster 5 (the most relevant variables)

```
> round(fkm6$Hraw[5,c(1,2,3,7,9,13,14)],2)
 Calories   Total Fat   Saturated Fat   Carbohydrates   Sugars   Iron (%DV)   Calcium(%DV)
     4.35        0.19            0.09            0.59     0.33         0.04           0.16
```

Oatmeal Raisin Cookie (the most relevant variables)

```
> round(fkm6$X[''Oatmeal Raisin Cookie'',c(1,2,3,7,9,13,14)],2)
 Calories   Total Fat   Saturated Fat   Carbohydrates   Sugars   Iron (%DV)   Calcium(%DV)
     4.55        0.18            0.08            0.67     0.39         0.06           0.18
```

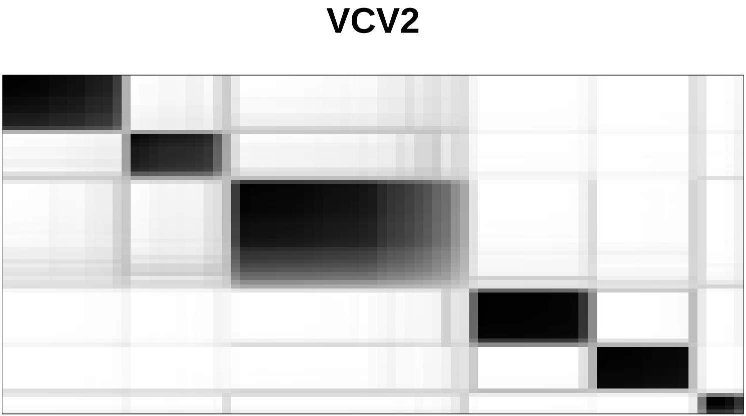
Baked Hot Apple Pie (the most relevant variables)

```
> round(fkm6$X[''Baked Hot Apple Pie'',c(1,2,3,7,9,13,14)],2)
 Calories   Total Fat   Saturated Fat   Carbohydrates   Sugars   Iron (%DV)   Calcium(%DV)
     3.25        0.17            0.09            0.42     0.17         0.03           0.08
```

Results Visualization: VCV2

Function `VCV2(Xca, U, which)`

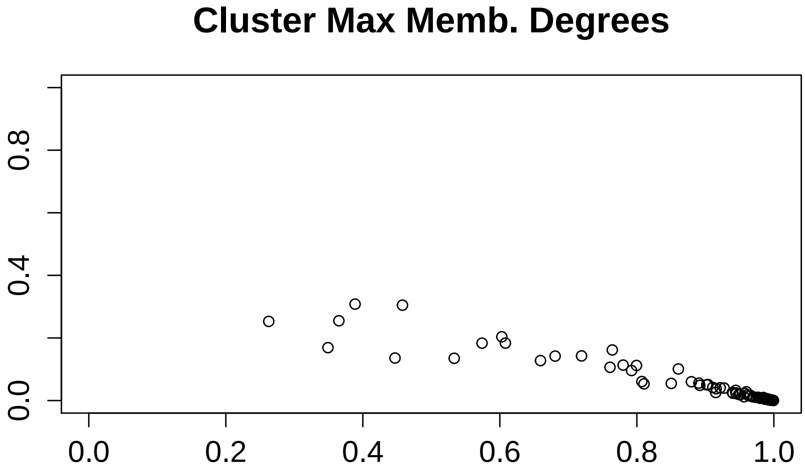
```
> VCV2(fkm6$Xca, fkm6$U, 2)
```



Results Visualization: VIFCR

Function `VIFCR(fclust.obj, which)`

```
> VIFCR(fkm6,2)
```



Unemployment data

The data set contains the unemployment rates and shares of 32 European countries in 2011 (source: Eurostat).

```
> library("fclust")
> data(unemployment)
```

Variables

✎ `numeric`:

- `Total.Rate`: the percentage of unemployed persons aged 15-74 in the economically active population
- `Youth.Rate`: the youth unemployment rate, defined as the unemployment rate for young people aged between 15 and 24
- `LongTerm.Share`: the long-term unemployment share, defined as the Percentage of unemployed persons who have been unemployed for 12 months or more

Aim of the analysis

Finding homogeneous groups of countries characterized by similar unemployment structures.

Correlation structure

$$Corr = \begin{bmatrix} 1 & 0.92 & 0.58 \\ 0.92 & 1 & 0.54 \\ 0.58 & 0.54 & 1 \end{bmatrix}$$

We decide to apply the Gustafson and Kessel extension of F*k*M (function `FKM.gk`) in order to explore the existence of clusters having non-spherical shapes.

FkM.gk with $k = 3$ clusters

Prior analyses on the data set suggest to run the algorithm using standardized data (`stand = 1`), and to choose $k = 3$ (`k = 3`) clusters (the default value $m = 2$ is set). The here-considered algorithm has a high risk of hitting local optima and, thus, 50 random starts are used (`RS = 50`).

```
> clust <- FKM.gk(unemployment, k = 3, RS =
50, stand = 1)
> cl.size(clust$U)
 Clus 1  Clus 2  Clus 3
     15       6      11
```

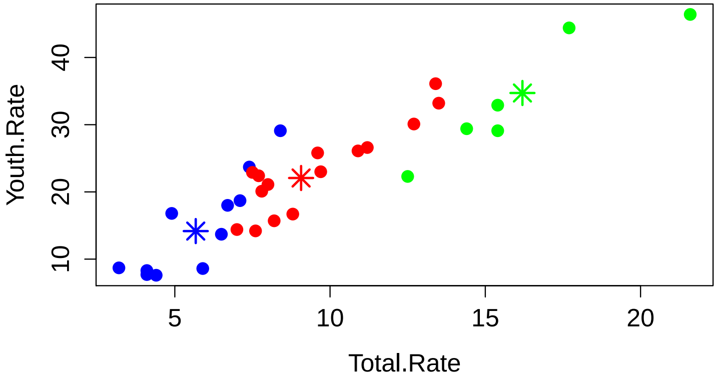
Clusters: covariance matrices

```
> clust$F
, , Clus 1
               Total.Rate Youth.Rate LongTerm.Share
 Total.Rate      1.299352   1.386309       2.770606
 Youth.Rate      1.386309   2.088642       2.875459
 LongTerm.Share  2.770606   2.875459       7.180983
, , Clus 2
               Total.Rate Youth.Rate LongTerm.Share
 Total.Rate      3.214435   3.511246      -1.801111
 Youth.Rate      3.511246   4.683005      -1.961230
 LongTerm.Share -1.801111  -1.961230       1.376300
, , Clus 3
               Total.Rate Youth.Rate LongTerm.Share
 Total.Rate      1.268973   1.859881       1.906008
 Youth.Rate      1.859881   3.822880       2.140836
 LongTerm.Share  1.906008   2.140836       3.969645
```

Results Visualization: plot.fclust (i)

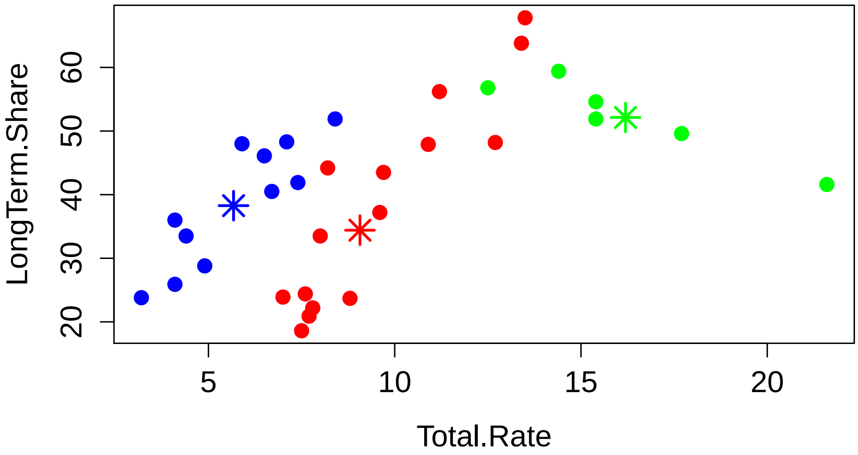
Function `plot.fclust(fclust.obj, v1v2, colclus, umin, ucex, pca)`

```
> plot.fclust(clust)
```



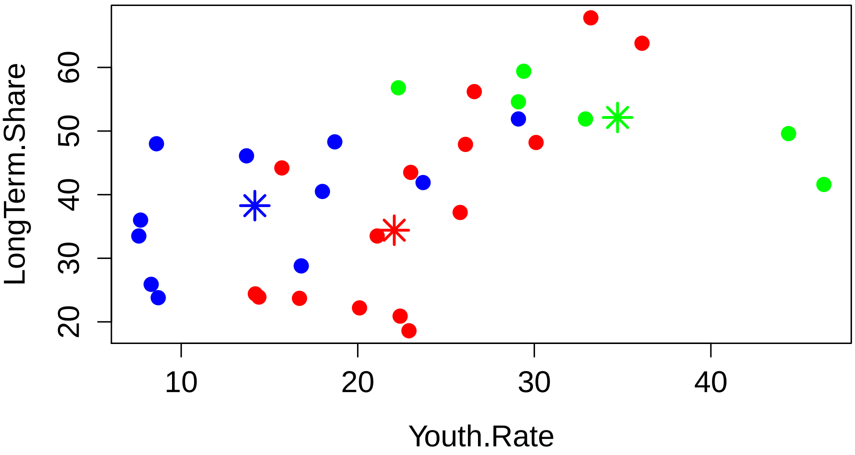
Results Visualization: plot.fclust (ii)

```
> plot.fclust(clust,v1v2=c(1,3))
```



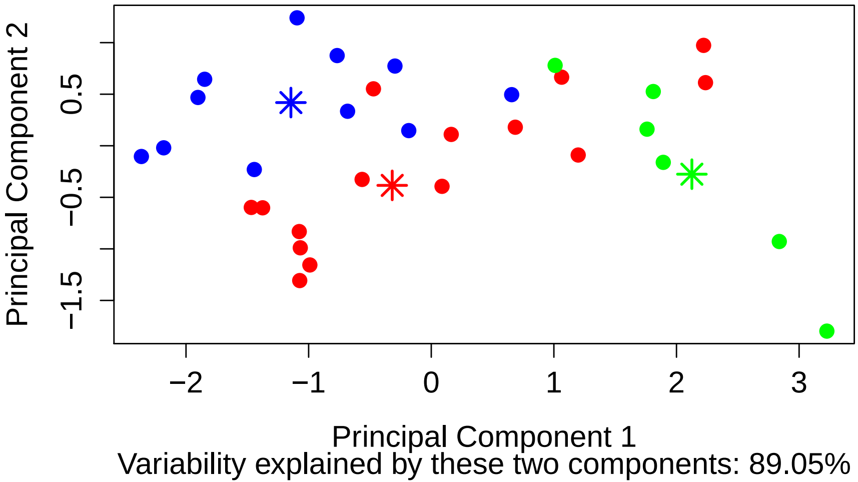
Results Visualization: plot.fclust (iii)

```
> plot.fclust(clust,v1v2=c(2,3))
```



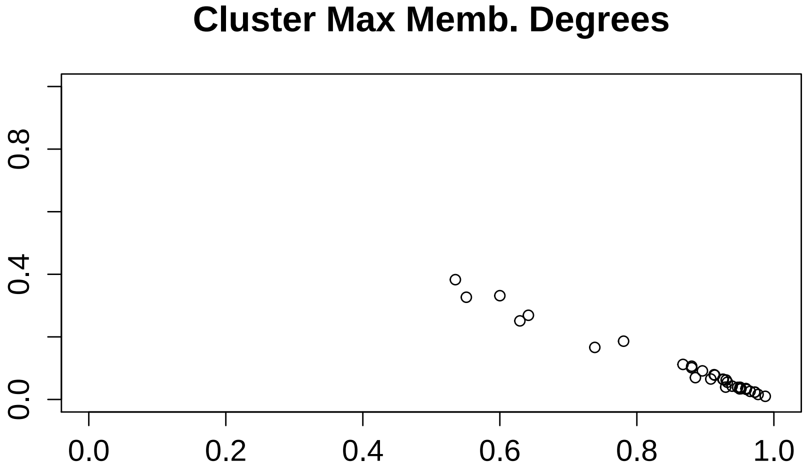
Results Visualization: plot.fclust (iv)

```
> plot.fclust(clust, pca=TRUE)
```



Results Visualization: VIFCR

```
> VIFCR(clust,2)
```



Clusters

Cluster 1: {Bulgaria, Croatia, Cyprus, Portugal, Denmark, Finland, France, Hungary, Iceland, Poland, Slovakia, Slovenia, Sweden, Turkey, UK}

Cluster 2: {Estonia, Ireland, Greece, Latvia, Lithuania, Spain}

Cluster 3: {Austria, Belgium, Czech Republic, Germany, Italy, Luxembourg, Malta, Netherlands, Norway, Romania, Switzerland}

Centroids

We now interpret the obtained clusters by studying the centroids (using the function `Hraw`) and the membership degree information.

```
> round(Hraw(clust$X, clust$H), 2)
        Total.Rate  Youth.Rate  LongTerm.Share
 Clus 1       9.07       22.07           34.41
 Clus 2      16.20       34.71           52.13
 Clus 3       5.67       14.17           38.27
```

Interpretation of the clusters

- Cluster 2 is composed by the Baltic states and a subset of the European countries mostly suffering from the economic crisis. Such a cluster is characterized by the highest levels of all the variables, hence highlighting a **critical situation**.
- By inspecting the centroids we can conclude that Cluster 1 detects countries with medium total and young unemployment rates and low long-term unemployment shares. Therefore, Cluster 1 seems to highlight **dynamic labour markets**.
- On the contrary, Cluster 3 represents **static labour markets**. In detail, it is composed by countries with low total and young unemployment rates and medium long-term unemployment share.

Fish data

Food balance sheet of Fish, year 2009 (FAO)

Variables

✎ `numeric`:

- `Production` (tonnes in live weight)
- `Imports` (tonnes in live weight)
- `Exports` (tonnes in live weight)
- `Population`: (thousands)
- `PCSupply`: Supply (kilograms per capita per year)
- `FishProtPC`: Fish Proteins (grams per capita per day)
- `AnimalProtPC`: Animal Proteins (grams per capita per day)
- `TotalProtPC`: Total Proteins (grams per capita per day)

units: 40 countries

Aim of the analysis

Finding homogeneous groups of countries characterized by similar behaviour related to production, imports and exports of fish, supply, fish, animal and total proteins.

- We have divided the first three variables by `Population`.
- We don't consider the variable `Population` in the cluster analysis.
- By inspecting the values of Fuzzy Silhouette for different number of clusters, it results that the optimal number is $k = 3$

FkM ($k = 3$ clusters)

Solution with $k = 3$ clusters (one low-size cluster)

```
> fkm    <-    FKM(X = fish, k = 3,
               m=2, stand = 1, RS = 10)

> cl.size(fkm$U)

  Cl 1  Cl 2  Cl 3
     2    20    18
```

✎ Cluster 1 contains Iceland and Faroe Island

Membership degrees

```
> round(fkm$clus[1:15,], 2)
             Cluster  Membership degree
 Albania           2               0.90
 Austria           3               0.59
 Belarus           2               0.91
 Belgium           3               0.70
 BosniaHerz        2               0.91
 Bulgaria          2               0.92
 Croatia           2               0.95
 CzechRep          2               0.92
 Denmark           3               0.64
 Estonia           2               0.79
 FaroeIs           1               0.96
 Finland           3               0.94
 FYRMacedonia      2               0.88
 France            3               0.97
 Germany           3               0.56
```

FkM with polynomial fuzzifier ($k = 3$ clusters)

Solution with $k = 3$ clusters (one low-size cluster)

```
> fkm.pf   <-   FKM.pf(X = fish, k = 3,
                b = 0.5, stand = 1, RS = 10)
> cl.size(fkm.pf$U)

  Cl 1  Cl 2  Cl 3
    19     2    19
```

- Cluster 2 contains Iceland and Faroe Island
- They seem to be noisy data

Membership degrees

```
> round(fkm.pf$clus[1:15,], 2)
              Cluster  Membership degree
 Albania            1               1.00
 Austria            3               0.92
 Belarus            1               1.00
 Belgium            3               1.00
 BosniaHerz         1               1.00
 Bulgaria           1               1.00
 Croatia            1               1.00
 CzechRep           1               1.00
 Denmark            3               1.00
 Estonia            1               1.00
 FaroeIs            2               1.00
 Finland            3               1.00
 FYRMacedonia       1               1.00
 France             3               1.00
 Germany            3               0.82
```

FkM with polynomial fuzzifier and noise clusters ($k = 2$ clusters)

Solution with $k = 2$ clusters

```
> fkm.pf.noise   <-   FKM.pf.noise(X = fish, k = 2,
                      b = 0.5, stand = 1, RS = 10)
```

Membership degrees (the most relevant countries)

```
> fkm.pf.noise$U
              Clus 1       Clus 2
Austria   0.28612643   0.67056640
FaroeIs   0.00000000   0.03307934
Germany   0.41432136   0.55093605
Iceland   0.00000000   0.29940487
Russian   0.66430809   0.30232321
```

Clusters

Cluster 1: {Albania, Belarusm, BosniaHerz, Bulgaria, Croatia, CzechRep, Estonia, FYRMacedonia, Hungary, Latvia, MoldovaRep, Montenegro, Poland, Romania, Russian, Serbia, Slovakia, Slovenia, Switzerland, Ukrainee}

Cluster 2: {Austria, Belgium, Denmark, Finland, France, Germany, Greece, Ireland, Italy, Lithuania, Luxemburg, Malta, Netherlands, Norway, Portugal, Spain, Sweden, UK }

Noise cluster: {FaroeIs, Iceland}

Mean values (the most relevant variables)

```
> apply(fkm.pf.noise$X[,c(1,2,3,4)],2,mean)
   Production  Imports  Exports  PCSupply
       456.55    35.34   261.54     23.39
```

Centroids (the most relevant variables)

```
> fkm.pf.noise$Hraw= Hraw(fkm.pf.noise$X,
fkm.pf.noise$H)
> round(fkm.pf.noise$Hraw[,c(1,2,3,4)],2)
          Production  Imports  Exports  PCSupply
 Clus 1        10.39    12.14    10.73     10.93
 Clus 2       139.06    48.14    99.53     31.36
```

FaroeIs (the most relevant variables)

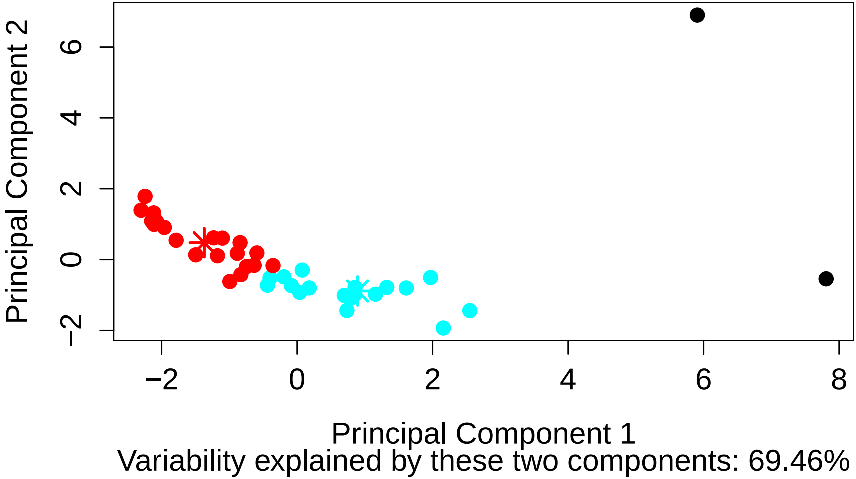
```
> round(fkm.pf.noise$X[``FaroeIs'',c(1,2,3,4)],2)
           Production  Imports  Exports  PCSupply
 FaroeIs     12491.59   115.47  6735.61     87.70
```

Iceland (the most relevant variables)

```
> round(fkm.pf.noise$X[``Iceland'',c(1,2,3,4)],2)
           Production  Imports  Exports  PCSupply
 Iceland      4443.25   240.46  2477.47     88.30
```

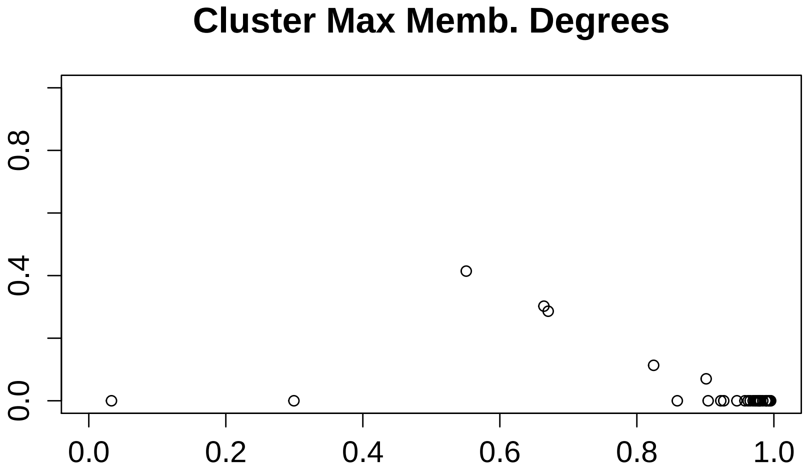
Results Visualization: plot.fclust

```
> plot.fclust(fkm.pf.noise, pca=TRUE)
```



Results Visualization: VIFCR

```
> VIFCR(fkm.pf.noise,2)
```



References

Bezdek JC. Journal of Cybernetics **3**, 58–73 (1974)

Bezdek JC. Pattern recognition with fuzzy objective function algorithm (1981)

Bezdek JC, Hathaway RJ. In: Proc. IJCNN 2002, IEEE Press, Piscataway, NJ, pp. 2225–2230 (2002).

Campello RJGB, Hruschka ER. Fuzzy Sets and Systems **157**, 2858–2875 (2006)

Davé RN. Pattern Recognition Letters **12**, 657–664 (1991)

Davé, RN. Pattern Recognition Letters **17**, 613–623 (1996)

Gustafson E, Kessel W. In: Proceedings of IEEE CDC (1978)

Hathaway RJ, Bezdek JC. Pattern Recognition Letters **24**, 1563–1569 (2003).

Huband JM, Bezdek JC. In: J.M. Zurada et al. (eds.), WCCI 2008, LNCS 5050, pp. 293–308 (2008).

Klawonn F, Höppner F. In: Advances in intelligent data analysis (LNCS 2779, pp. 254-264) (2003).

Klawonn F, Chekhtman V, Janz E. In: J. Benitez, O. Cordon, F. Hoffmann, R. Roy (eds.): Advances in Soft Computing: Engineering Design and Manufacturing. Springer, London, 65–76 (2003).

Kaufman L, Rousseeuw PJ. In: Statistical Data Analysis Based on the L1 -norm and Related Methods, 405-416 (1987)

Kaufman L, Rousseeuw PJ. Finding Groups in Data: An Introduction to Cluster Analysis, J. Wiley and Sons (1990)

Li RP, Mukaidono M. In: Proceedings of FUZZ-IEEE/IFES 95, pp. 2227–2232 (1995)

Li RP, Mukaidono M. Fuzzy Sets and Systems **102**, 253–258 (1999)

Thank you!

e-mail: **mariabrigida.ferraro@uniroma1.it**