Package: fclust (via r-universe)

November 6, 2024

Type Package

Title Fuzzy Clustering

Version 2.1.1.1

Date 2019-09-16

Author Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

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 (>= 3.3), base, stats, graphics, grDevices, utils

Imports Rcpp (>= 0.12.5), MASS (>= 7.3)

LinkingTo Rcpp, RcppArmadillo (>= 0.7)

License GPL (>= 2)

ByteCompile true

Repository CRAN

NeedsCompilation yes

LazyLoad yes

Encoding UTF-8

Date/Publication 2022-11-16 16:34:44 UTC

Contents

ARI.F . . . 3
butterfly . . . 4
cl.memb . . . 5
cl.memb.H . . . 6
cl.memb.t . . . 7
cl.size . . . 8
cl.size.H . . . 9
Fclust . . . 10

Fclust.compare 11
Fclust.index 13
FKM 14
FKM.ent 16
FKM.ent.noise 18
FKM.gk 20
FKM.gk.ent 23
FKM.gk.ent.noise 25
FKM.gk.noise 28
FKM.gkb 30
FKM.gkb.ent 32
FKM.gkb.ent.noise 34
FKM.gkb.noise 37
FKM.med 39
FKM.med.noise 42
FKM.noise 44
FKM.pf 46
FKM.pf.noise 48
houseVotes 50
Hraw 51
JACCARD.F 52
Mc 54
MPC 55
NBA 56
NEFRC 57
NEFRC.noise 59
PC 61
PE 62
plot.fclust 63
print.fclust 65
RI.F 66
SIL 67
SIL.F 68
summary.fclust 69
synt.data 70
synt.data2 71
unemployment 72
VAT 73
VCV 74
VCV2 75
VIFCR 77
XB 78

Index **80**

ARI.F *Fuzzy adjusted Rand index*

Description

Produces the fuzzy version of the adjusted Rand index between a hard (reference) partition and a fuzzy partition.

Usage

```
ARI.F(VC, U, t_norm)
```

Arguments

`VC`	Vector of class labels
`U`	Fuzzy membership degree matrix or data.frame
`t_norm`	Type of the triangular norm: "minimum" (minimum triangular norm), "triangular product" (product norm) (default: "minimum")

Value

`ari.f`Value of the fuzzy adjusted Rand index

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Campello, R.J., 2007. A fuzzy extension of the Rand index and other related indexes for clustering and classification assessment. Pattern Recognition Letters, 28, 833-841.
Hubert, L., Arabie, P., 1985. Comparing partitions. Journal of Classification, 2, 193-218.

See Also

`RI.F`, `JACCARD.F`, `Fclust.compare`

Examples

```
## Not run:
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
```

```
## fuzzy k-means
## (excluded the factor column Type (last column))
clust=FKM(Mc[,1:(ncol(Mc)-1)],k=6,m=1.5,stand=1)
## fuzzy adjusted Rand index
ari.f=ARI.F(VC=Mc$Type,U=clust$U)

## End(Not run)
```

---

butterfly *Butterfly data*

Description

Synthetic dataset with 2 clusters and some outliers.

Usage

```
data(butterfly)
```

Format

A matrix with 17 rows and 2 columns.

Details

The butterfly data motivate the need for the fuzzy approach to clustering.
The presence of outliers can be handled using fuzzy *k*-means with noise cluster. In fact, differently from fuzzy *k*-means, the membership degrees of the outliers are low for all the clusters.

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

See Also

Fclust, FKM, FKM.noise

Examples

```
## butterfly data
data(butterfly)
plot(butterfly,type='n')
text(butterfly[,1],butterfly[,2],labels=rownames(butterfly),cex=0.7,lwd=2)
## membership degree matrix using fuzzy k-means (rounded)
round(FKM(butterfly)$U,2)
## membership degree matrix using fuzzy k-means with noise cluster (rounded)
round(FKM.noise(butterfly,delta=3)$U,2)
```

cl.memb *Cluster membership*

Description

Produces a summary of the membership degree information.

Usage

```
cl.memb (U)
```

Arguments

U	Membership degree matrix

Details

An object is assigned to a cluster according to the maximal membership degree. Therefore, it produces the closest hard clustering partition

Value

info.U	Matrix containing the indexes of the clusters where the objects are assigned (row 1) and the associated membership degrees (row 2)

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

See Also

cl.memb.H, cl.memb.t

Examples

```
n=20
k=3
## randomly generated membership degree matrix
U=matrix(runif(n*k,0,1), nrow=n, ncol=k)
U=U/apply(U,1,sum)
info.U=cl.memb(U)
## objects assigned to cluster 2
rownames(info.U[info.U[,1]==2,])
```

`cl.memb.H` *Cluster membership*

Description

Produces a summary of the membership degree information in the hard clustering sense (objects are considered to be assigned to clusters only if the corresponding membership degree are >=0.5).

Usage

```
cl.memb.H (U)
```

Arguments

`U`	Membership degree matrix

Details

An object is assigned to a cluster according to the maximal membership degree provided that such a maximal membership degree is >=0.5, otherwise it is assumed that an object is not assigned to any cluster (denoted by cluster index = 0 in row 1).

Value

`info.U`	Matrix containing the indexes of the clusters where the objects are assigned (row 1) and the associated membership degrees (row 2)

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

See Also

`cl.memb`, `cl.memb.t`

Examples

```
n=20
k=3
## randomly generated membership degree matrix
U=matrix(runif(n*k,0,1), nrow=n, ncol=k)
U=U/apply(U,1,sum)
info.U=cl.memb.H(U)
## objects assigned to clusters in the hard clustering sense
rownames(info.U[info.U[,1]!=0,])
```

`cl.memb.t` *Cluster membership*

Description

Produces a summary of the membership degree information according to a threshold.

Usage

```
cl.memb.t (U, t)
```

Arguments

`U`	Membership degree matrix
`t`	Threshold in [0,1] (default: 0)

Details

An object is assigned to a cluster according to the maximal membership degree provided that such a maximal membership degree is >= `t`, otherwise it is assumed that an object is not assigned to any cluster (denoted by cluster index = 0 in row 1). The function can be useful to select the subset of objects clearly assigned to clusters (objects with maximal membership degrees >= `t`).

Value

`info.U`	Matrix containing the indexes of the clusters where the objects are assigned (row 1) and the associated membership degrees (row 2)

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

See Also

`cl.memb`, `cl.memb.H`

Examples

```
n=20
k=3
## randomly generated membership degree matrix
U=matrix(runif(n*k,0,1), nrow=n, ncol=k)
U=U/apply(U,1,sum)
## threshold t=0.6
info.U=cl.memb.t(U,0.6)
## objects clearly assigned to clusters
rownames(info.U[info.U[,1]!=0,])
```

cl.size *Cluster size*

Description

Produces the sizes of the clusters.

Usage

```
cl.size (U)
```

Arguments

U	Membership degree matrix

Details

An object is assigned to a cluster according to the maximal membership degree.

Value

clus.size	Vector containing the sizes of the clusters

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

See Also

cl.size.H

Examples

```
n=20
k=3
## randomly generated membership degree matrix
U=matrix(runif(n*k,0,1), nrow=n, ncol=k)
U=U/apply(U,1,sum)
clus.size=cl.size(U)
```

cl.size.H *Cluster size*

Description

Produces the sizes of the clusters in the hard clustering sense (objects are considered to be assigned to clusters only if the corresponding membership degree are >=0.5).

Usage

```
cl.size.H (U)
```

Arguments

U	Membership degree matrix

Details

An object is assigned to a cluster according to the maximal membership degree provided that such a maximal membership degree is >=0.5, otherwise it is assumed that an object is not assigned to any cluster.

Value

clus.size	Vector containing the sizes of the clusters

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

See Also

cl.size

Examples

```
n=20
k=3
## randomly generated membership degree matrix
U=matrix(runif(n*k,0,1), nrow=n, ncol=k)
U=U/apply(U,1,sum)
## cluster size in the hard clustering sense
clus.size=cl.size.H(U)
```

Fclust *Fuzzy clustering*

Description

Performs fuzzy clustering by using the algorithms available in the package.

Usage

```
Fclust (X, k, type, ent, noise, stand, distance)
```

Arguments

X	Matrix or data.frame
k	An integer value specifying the number of clusters (default: 2)
type	Fuzzy clustering algorithm: "standard" (standard algorithms: FKM - type if distance=TRUE, NEFRC - type if if distance=FALSE), "polynomial" (algorithms with the polynomial fuzzifier), "gk" (Gustafson and Kessel - like algorithms), "gkb" (Gustafson, Kessel and Babuska - like algorithms), "medoids" (Medoid - based algorithms) (default: "standard")
ent	If ent=TRUE, the entropy regularization variant of the algorithm is run (default: FALSE)
noise	If noise=TRUE, the noise cluster variant of the algorithm is run (default: FALSE)
stand	Standardization: if stand=1, the clustering algorithm is run using standardized data (default: no standardization)
distance	If distance=TRUE, X is assumed to be a distance/dissimilarity matrix (default: FALSE)

Details

The clustering algorithms are run by using default options.
To specify different options, use the corresponding function.

Value

clust	Object of class fclust

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

See Also

print.fclust, summary.fclust, plot.fclust, FKM, FKM.ent, FKM.gk, FKM.gk.ent, FKM.gkb, FKM.gkb.ent, FKM.med, FKM.pf, FKM.noise, FKM.ent.noise, FKM.gk.noise, FKM.gkb.ent.noise, FKM.gkb.noise, FKM.gk.ent.noise,FKM.med.noise, FKM.pf.noise, NEFRC, NEFRC.noise, Fclust.index, Fclust.compare

Examples

```
## Not run:
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
## fuzzy k-means
## (excluded the factor column Type (last column))
clust=Fclust(Mc[,1:(ncol(Mc)-1)],k=6,type="standard",ent=FALSE,noise=FALSE,stand=1,distance=FALSE)
## fuzzy k-means with polynomial fuzzifier
## (excluded the factor column Type (last column))
clust=Fclust(Mc[,1:(ncol(Mc)-1)],k=6,type="polynomial",ent=FALSE,noise=FALSE,stand=1,distance=FALSE)
## fuzzy k-means with entropy regularization
## (excluded the factor column Type (last column))
clust=Fclust(Mc[,1:(ncol(Mc)-1)],k=6,type="standard",ent=TRUE,noise=FALSE,stand=1,distance=FALSE)
## fuzzy k-means with noise cluster
## (excluded the factor column Type (last column))
clust=Fclust(Mc[,1:(ncol(Mc)-1)],k=6,type="standard",ent=FALSE,noise=TRUE,stand=1,distance=FALSE)

## End(Not run)
```

`Fclust.compare` *Similarity between partitions*

Description

Performs some measures of similarity between a hard (reference) partition and a fuzzy partition.

Usage

```
Fclust.compare(VC, U, index, tnorm)
```

Arguments

`VC`	Vector of class labels
`U`	Fuzzy membership degree matrix or data.frame
`index`	Measures of similarity: "ARI.F" (fuzzy version of the adjuster Rand index), "RI.F" (fuzzy version of the Rand index), "JACCARD.F" (fuzzy version of the Jaccard index), "ALL" for all the indexes (default: "ALL")
`tnorm`	Type of the triangular norm: "minimum" (minimum triangular norm), "triangular product" (product norm) (default: "minimum")

Details

`index` is not case-sensitive. All the measures of similarity share the same properties of their non-fuzzy counterpart.

Value

`out.index`Vector containing the similarity measures

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Campello, R.J., 2007. A fuzzy extension of the Rand index and other related indexes for clustering and classification assessment. Pattern Recognition Letters, 28, 833-841.
Hubert, L., Arabie, P., 1985. Comparing partitions. Journal of Classification, 2, 193-218.
Jaccard, P., 1901. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin de la Société Vaudoise des Sciences Naturelles, 37, 547-579.
Rand, W.M., 1971. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association, 66, 846-850.

See Also

`RI.F`, `ARI.F`, `JACCARD.F`

Examples

```
## Not run:
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
## fuzzy k-means
## (excluded the factor column Type (last column))
clust=FKM(Mc[,1:(ncol(Mc)-1)],k=6,m=1.5,stand=1)
## all measures of similarity
all.indexes=Fclust.compare(VC=Mc$Type,U=clust$U)
## fuzzy adjusted Rand index
Fari.index=Fclust.compare(VC=Mc$Type,U=clust$U,index="ARI.F")

## End(Not run)
```

Fclust.index *Cluster validity indexes*

Description

Performs some cluster validity indexes for choosing the optimal number of clusters *k*.

Usage

```
Fclust.index (fclust.obj, index, alpha)
```

Arguments

fclust.obj	Object of class fclust
index	Cluster validity indexes to select the number of clusters: PC (partition coefficient), PE (partition entropy), MPC (modified partition coefficient), SIL (silhouette), SIL.F (fuzzy silhouette), XB (Xie and Beni),ALL for all the indexes (default: "ALL")
alpha	Weighting coefficient for the fuzzy silhouette index SIL.F (default: 1)

Details

index is not case-sensitive.

Value

out.index	Vector containing the index values

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

See Also

PC, PE, MPC, SIL, SIL.F, XB, Fclust, Mc

Examples

```
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
## fuzzy k-means
## (excluded the factor column Type (last column))
```

```
clust=FKM(Mc[,1:(ncol(Mc)-1)],k=6,m=1.5,stand=1)
## cluster validity indexes
all.indexes=Fclust.index(clust)
## Xie and Beni cluster validity index
XB.index=Fclust.index(clust,'XB')
```

FKM *Fuzzy k-means*

Description

Performs the fuzzy *k*-means clustering algorithm.

Usage

```
FKM (X, k, m, RS, stand, startU, index, alpha, conv, maxit, seed)
```

Arguments

`X`	Matrix or data.frame
`k`	An integer value or vector specifying the number of clusters for which the `index` is to be calculated (default: 2:6)
`m`	Parameter of fuzziness (default: 2)
`RS`	Number of (random) starts (default: 1)
`stand`	Standardization: if `stand=1`, the clustering algorithm is run using standardized data (default: no standardization)
`startU`	Rational start for the membership degree matrix `U` (default: no rational start)
`index`	Cluster validity index to select the number of clusters: `"PC"` (partition coefficient), `"PE"` (partition entropy), `"MPC"` (modified partition coefficient), `"SIL"` (silhouette), `"SIL.F"` (fuzzy silhouette), `"XB"` (Xie and Beni) (default: "SIL.F")
`alpha`	Weighting coefficient for the fuzzy silhouette index `SIL.F` (default: 1)
`conv`	Convergence criterion (default: 1e-9)
`maxit`	Maximum number of iterations (default: 1e+6)
`seed`	Seed value for random number generation (default: NULL)

Details

If `startU` is given, the argument `k` is ignored (the number of clusters is `ncol(startU)`).
If `startU` is given, the first element of `value`, `cput` and `iter` refer to the rational start.

Value

Object of class `fclust`, which is a list with the following components:

`U`	Membership degree matrix
`H`	Prototype matrix
`F`	Array containing the covariance matrices of all the clusters (`NULL` for `FKM`)
`clus`	Matrix containing the indexes of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
`medoid`	Vector containing the indexes of the medoid objects (`NULL` for `FKM`)
`value`	Vector containing the loss function values for the `RS` starts
`criterion`	Vector containing the values of the cluster validity index
`iter`	Vector containing the numbers of iterations for the `RS` starts
`k`	Number of clusters
`m`	Parameter of fuzziness
`ent`	Degree of fuzzy entropy (`NULL` for `FKM`)
`b`	Parameter of the polynomial fuzzifier (`NULL` for `FKM`)
`vp`	Volume parameter (`NULL` for `FKM`)
`delta`	Noise distance (`NULL` for `FKM`)
`gam`	Weighting parameter for the fuzzy covariance matrices (`NULL` for `FKM`)
`mcn`	Maximum condition number for the fuzzy covariance matrices (`NULL` for `FKM`)
`stand`	Standardization (Yes if `stand=1`, No if `stand=0`)
`Xca`	Data used in the clustering algorithm (standardized data if `stand=1`)
`X`	Raw data
`D`	Dissimilarity matrix (`NULL` for `FKM`)
`call`	Matched call

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Bezdek J.C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York.

See Also

`FKM.noise`, `Fclust`, `Fclust.index`, `print.fclust`, `summary.fclust`, `plot.fclust`, `Mc`

Examples

```
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
## fuzzy k-means (excluded the factor column Type (last column)), fixing the number of clusters
clust=FKM(Mc[,1:(ncol(Mc)-1)],k=6,m=1.5,stand=1)
## fuzzy k-means (excluded the factor column Type (last column)), selecting the number of clusters
clust=FKM(Mc[,1:(ncol(Mc)-1)],k=2:6,m=1.5,stand=1)
```

`FKM.ent` *Fuzzy k-means with entropy regularization*

Description

Performs the fuzzy k-means clustering algorithm with entropy regularization.
The entropy regularization allows us to avoid using the artificial fuzziness parameter m. This is replaced by the degree of fuzzy entropy *ent*, related to the concept of temperature in statistical physics. An interesting property of the fuzzy k-means with entropy regularization is that the prototypes are obtained as weighted means with weights equal to the membership degrees (rather than to the membership degrees at the power of m as is for the fuzzy k-means).

Usage

```
FKM.ent (X, k, ent, RS, stand, startU, index, alpha, conv, maxit, seed)
```

Arguments

`X`	Matrix or data.frame
`k`	An integer value or vector specifying the number of clusters for which the `index` is to be calculated (default: 2:6)
`ent`	Degree of fuzzy entropy (default: 1)
`RS`	Number of (random) starts (default: 1)
`stand`	Standardization: if `stand=1`, the clustering algorithm is run using standardized data (default: no standardization)
`startU`	Rational start for the membership degree matrix `U` (default: no rational start)
`index`	Cluster validity index to select the number of clusters: `"PC"` (partition coefficient), `"PE"` (partition entropy), `"MPC"` (modified partition coefficient), `"SIL"` (silhouette), `"SIL.F"` (fuzzy silhouette), `"XB"` (Xie and Beni) (default: "SIL.F")
`alpha`	Weighting coefficient for the fuzzy silhouette index `SIL.F` (default: 1)
`conv`	Convergence criterion (default: 1e-9)
`maxit`	Maximum number of iterations (default: 1e+6)
`seed`	Seed value for random number generation (default: NULL)

Details

If startU is given, the argument k is ignored (the number of clusters is ncol(startU)).
If startU is given, the first element of value, cput and iter refer to the rational start.
The default value for ent is in general not reasonable if FKM.ent is run using raw data.
The update of the membership degrees requires the computation of exponential functions. In some cases, this may produce NaN values and the algorithm stops. Such a problem is usually solved by running FKM.ent using standardized data (stand=1).

Value

Object of class fclust, which is a list with the following components:

U	Membership degree matrix
H	Prototype matrix
F	Array containing the covariance matrices of all the clusters (NULL for FKM.ent)
clus	Matrix containing the indexes of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
medoid	Vector containing the indexes of the medoid objects (NULL for FKM.ent)
value	Vector containing the loss function values for the RS starts
criterion	Vector containing the values of the cluster validity index
iter	Vector containing the numbers of iterations for the RS starts
k	Number of clusters
m	Parameter of fuzziness (NULL for FKM.ent)
ent	Degree of fuzzy entropy
b	Parameter of the polynomial fuzzifier (NULL for FKM.ent)
vp	Volume parameter (NULL for FKM.ent)
delta	Noise distance (NULL for FKM.ent)
gam	Weighting parameter for the fuzzy covariance matrices (NULL for FKM.ent)
mcn	Maximum condition number for the fuzzy covariance matrices (NULL for FKM.ent)
stand	Standardization (Yes if stand=1, No if stand=0)
Xca	Data used in the clustering algorithm (standardized data if stand=1)
X	Raw data
D	Dissimilarity matrix (NULL for FKM.ent)
call	Matched call

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Li R., Mukaidono M., 1995. A maximum entropy approach to fuzzy clustering. Proceedings of the Fourth IEEE Conference on Fuzzy Systems (FUZZ-IEEE/IFES '95), pp. 2227-2232.
Li R., Mukaidono M., 1999. Gaussian clustering method based on maximum-fuzzy-entropy interpretation. Fuzzy Sets and Systems, 102, 253-258.

See Also

FKM.ent.noise, Fclust, Fclust.index, print.fclust, summary.fclust, plot.fclust, Mc

Examples

```
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
## fuzzy k-means with entropy regularization, fixing the number of clusters
## (excluded the factor column Type (last column))
clust=FKM.ent(Mc[,1:(ncol(Mc)-1)],k=6,ent=3,RS=10,stand=1)
## fuzzy k-means with entropy regularization, selecting the number of clusters
## (excluded the factor column Type (last column))
clust=FKM.ent(Mc[,1:(ncol(Mc)-1)],k=2:6,ent=3,RS=10,stand=1)
```

FKM.ent.noise *Fuzzy k-means with entropy regularization and noise cluster*

Description

Performs the fuzzy *k*-means clustering algorithm with entropy regularization and noise cluster.
The entropy regularization allows us to avoid using the artificial fuzziness parameter *m*. This is replaced by the degree of fuzzy entropy *ent*, related to the concept of temperature in statistical physics. An interesting property of the fuzzy *k*-means with entropy regularization is that the prototypes are obtained as weighted means with weights equal to the membership degrees (rather than to the membership degrees at the power of *m* as is for the fuzzy k-means).
The noise cluster is an additional cluster (with respect to the *k* standard clusters) such that objects recognized to be outliers are assigned to it with high membership degrees.

Usage

```
FKM.ent.noise (X, k, ent, delta, RS, stand, startU, index, alpha, conv, maxit, seed)
```

Arguments

`X`	Matrix or data.frame
`k`	An integer value or vector specifying the number of clusters for which the `index` is to be calculated (default: 2:6)
`ent`	Degree of fuzzy entropy (default: 1)
`delta`	Noise distance (default: average Euclidean distance between objects and prototypes from `FKM.ent` using the same values of `k` and `m`)
`RS`	Number of (random) starts (default: 1)

stand	Standardization: if stand=1, the clustering algorithm is run using standardized data (default: no standardization)
startU	Rational start for the membership degree matrix U (default: no rational start)
index	Cluster validity index to select the number of clusters: "PC" (partition coefficient), "PE" (partition entropy), "MPC" (modified partition coefficient), "SIL" (silhouette), "SIL.F" (fuzzy silhouette), "XB" (Xie and Beni) (default: "SIL.F")
alpha	Weighting coefficient for the fuzzy silhouette index SIL.F (default: 1)
conv	Convergence criterion (default: 1e-9)
maxit	Maximum number of iterations (default: 1e+6)
seed	Seed value for random number generation (default: NULL)

Details

If startU is given, the argument k is ignored (the number of clusters is ncol(startU)).
If startU is given, the first element of value, cput and iter refer to the rational start.
The default value for ent is in general not reasonable if FKM.ent is run using raw data.
The update of the membership degrees requires the computation of exponential functions. In some cases, this may produce NaN values and the algorithm stops. Such a problem is usually solved by running FKM.ent using standardized data (stand=1).

Value

Object of class fclust, which is a list with the following components:

U	Membership degree matrix
H	Prototype matrix
F	Array containing the covariance matrices of all the clusters (NULL for FKM.ent.noise)
clus	Matrix containing the indexes of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
medoid	Vector containing the indexes of the medoid objects (NULL for FKM.ent.noise)
value	Vector containing the loss function values for the RS starts
criterion	Vector containing the values of the cluster validity index
iter	Vector containing the numbers of iterations for the RS starts
k	Number of clusters
m	Parameter of fuzziness (NULL for FKM.ent.noise)
ent	Degree of fuzzy entropy
b	Parameter of the polynomial fuzzifier (NULL for FKM.ent.noise)
vp	Volume parameter (NULL for FKM.ent.noise)
delta	Noise distance
gam	Weighting parameter for the fuzzy covariance matrices (NULL for FKM.ent.noise)
mcn	Maximum condition number for the fuzzy covariance matrices (NULL for FKM.ent.noise)
stand	Standardization (Yes if stand=1, No if stand=0)

Xca	Data used in the clustering algorithm (standardized data if stand=1)
X	Raw data
D	Dissimilarity matrix (NULL for FKM.ent.noise)
call	Matched call

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Dave' R.N., 1991. Characterization and detection of noise in clustering. Pattern Recognition Letters, 12, 657-664.
Li R., Mukaidono M., 1995. A maximum entropy approach to fuzzy clustering. Proceedings of the Fourth IEEE Conference on Fuzzy Systems (FUZZ-IEEE/IFES '95), pp. 2227-2232.
Li R., Mukaidono M., 1999. Gaussian clustering method based on maximum-fuzzy-entropy interpretation. Fuzzy Sets and Systems, 102, 253-258.

See Also

FKM.ent, Fclust, Fclust.index, print.fclust, summary.fclust, plot.fclust, butterfly

Examples

```
## butterfly data
data(butterfly)
## fuzzy k-means with entropy regularization and noise cluster, fixing the number of clusters
clust=FKM.ent.noise(butterfly,k = 2, RS=5,delta=3)
## fuzzy k-means with entropy regularization and noise cluster, selecting the number of clusters
clust=FKM.ent.noise(butterfly,RS=5,delta=3)
```

FKM.gk *Gustafson and Kessel - like fuzzy k-means*

Description

Performs the Gustafson and Kessel - like fuzzy k-means clustering algorithm.
Differently from fuzzy k-means, it is able to discover non-spherical clusters.

Usage

```
FKM.gk (X, k, m, vp, RS, stand, startU, index, alpha, conv, maxit, seed)
```

Arguments

X	Matrix or data.frame
k	An integer value or vector specifying the number of clusters for which the index is to be calculated (default: 2:6)
m	Parameter of fuzziness (default: 2)
vp	Volume parameter (default: rep(1,k))
RS	Number of (random) starts (default: 1)
stand	Standardization: if stand=1, the clustering algorithm is run using standardized data (default: no standardization)
startU	Rational start for the membership degree matrix U (default: no rational start)
index	Cluster validity index to select the number of clusters: "PC" (partition coefficient), "PE" (partition entropy), "MPC" (modified partition coefficient), "SIL" (silhouette), "SIL.F" (fuzzy silhouette), "XB" (Xie and Beni) (default: "SIL.F")
alpha	Weighting coefficient for the fuzzy silhouette index SIL.F (default: 1)
conv	Convergence criterion (default: 1e-9)
maxit	Maximum number of iterations (default: 1e+6)
seed	Seed value for random number generation (default: NULL)

Details

If startU is given, the argument k is ignored (the number of clusters is ncol(startU)).
If startU is given, the first element of value, cput and iter refer to the rational start.
If a cluster covariance matrix becomes singular, then the algorithm stops and the element of value is NaN.
The Babuska et al. variant in FKM.gkb is recommended.

Value

Object of class fclust, which is a list with the following components:

U	Membership degree matrix
H	Prototype matrix
F	Array containing the covariance matrices of all the clusters
clus	Matrix containing the indexes of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
medoid	Vector containing the indexes of the medoid objects (NULL for FKM.gk)
value	Vector containing the loss function values for the RS starts
criterion	Vector containing the values of the cluster validity index
iter	Vector containing the numbers of iterations for the RS starts
k	Number of clusters
m	Parameter of fuzziness
ent	Degree of fuzzy entropy (NULL for FKM.gk)

b	Parameter of the polynomial fuzzifier (NULL for FKM.gk)
vp	Volume parameter (default: rep(1,max(k)). If k is a vector, for each group the first k element of vpare considered.
delta	Noise distance (NULL for FKM.gk)
gam	Weighting parameter for the fuzzy covariance matrices (NULL for FKM.gk)
mcn	Maximum condition number for the fuzzy covariance matrices (NULL for FKM.gk)
stand	Standardization (Yes if stand=1, No if stand=0)
Xca	Data used in the clustering algorithm (standardized data if stand=1)
X	Raw data
D	Dissimilarity matrix (NULL for FKM.gk)
call	Matched call

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Gustafson E.E., Kessel W.C., 1978. Fuzzy clustering with a fuzzy covariance matrix. Proceedings of the IEEE Conference on Decision and Control, pp. 761-766.

See Also

FKM.gkb, Fclust, Fclust.index, print.fclust, summary.fclust, plot.fclust, unemployment

Examples

```
## Not run:
## unemployment data
data(unemployment)
## Gustafson and Kessel-like fuzzy k-means, fixing the number of clusters
clust=FKM.gk(unemployment,k=3,RS=10)
## Gustafson and Kessel-like fuzzy k-means, selecting the number of clusters
clust=FKM.gk(unemployment,k=2:6,RS=10)

## End(Not run)
```

FKM.gk.ent *Gustafson and Kessel - like fuzzy k-means with entropy regularization*

Description

Performs the Gustafson and Kessel - like fuzzy *k*-means clustering algorithm with entropy regularization.
Differently from fuzzy *k*-means, it is able to discover non-spherical clusters.
The entropy regularization allows us to avoid using the artificial fuzziness parameter *m*. This is replaced by the degree of fuzzy entropy *ent*, related to the concept of temperature in statistical physics. An interesting property of the fuzzy *k*-means with entropy regularization is that the prototypes are obtained as weighted means with weights equal to the membership degrees (rather than to the membership degrees at the power of *m* as is for the fuzzy k-means).

Usage

```
FKM.gk.ent (X, k, ent, vp, RS, stand, startU, index, alpha, conv, maxit, seed)
```

Arguments

`X`	Matrix or data.frame
`k`	An integer value or vector specifying the number of clusters for which the `index` is to be calculated (default: 2:6)
`ent`	Degree of fuzzy entropy (default: 1)
`vp`	Volume parameter (default: rep(1,k))
`RS`	Number of (random) starts (default: 1)
`stand`	Standardization: if `stand=1`, the clustering algorithm is run using standardized data (default: no standardization)
`startU`	Rational start for the membership degree matrix `U` (default: no rational start)
`index`	Cluster validity index to select the number of clusters: `"PC"` (partition coefficient), `"PE"` (partition entropy), `"MPC"` (modified partition coefficient), `"SIL"` (silhouette), `"SIL.F"` (fuzzy silhouette), `"XB"` (Xie and Beni) (default: "SIL.F")
`alpha`	Weighting coefficient for the fuzzy silhouette index `SIL.F` (default: 1)
`conv`	Convergence criterion (default: 1e-9)
`maxit`	Maximum number of iterations (default: 1e+6)
`seed`	Seed value for random number generation (default: NULL)

Details

If `startU` is given, the argument `k` is ignored (the number of clusters is `ncol(startU)`).
If `startU` is given, the first element of `value`, `cput` and `iter` refer to the rational start.
If a cluster covariance matrix becomes singular, the algorithm stops and the element of `value` is NaN.
The default value for `ent` is in general not reasonable if `FKM.gk.ent` is run using raw data.

The update of the membership degrees requires the computation of exponential functions. In some cases, this may produce NaN values and the algorithm stops. Such a problem is usually solved by running FKM.gk.ent using standardized data (stand=1).
The Babuska et al. variant in FKM.gkb.ent is recommended.

Value

Object of class fclust, which is a list with the following components:

U	Membership degree matrix
H	Prototype matrix
F	Array containing the covariance matrices of all the clusters
clus	Matrix containing the indexes of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
medoid	Vector containing the indexes of the medoid objects (NULL for FKM.gk.ent)
value	Vector containing the loss function values for the RS starts
criterion	Vector containing the values of the cluster validity index
iter	Vector containing the numbers of iterations for the RS starts
k	Number of clusters
m	Parameter of fuzziness (NULL for FKM.gk.ent)
ent	Degree of fuzzy entropy
b	Parameter of the polynomial fuzzifier (NULL for FKM.gk.ent)
vp	Volume parameter (default: rep(1,max(k)). If k is a vector, for each group the first k element of vpare considered.
delta	Noise distance (NULL for FKM.gk.ent)
gam	Weighting parameter for the fuzzy covariance matrices (NULL for FKM.gk.ent)
mcn	Maximum condition number for the fuzzy covariance matrices (NULL for FKM.gk.ent)
stand	Standardization (Yes if stand=1, No if stand=0)
Xca	Data used in the clustering algorithm (standardized data if stand=1)
X	Raw data
D	Dissimilarity matrix (NULL for FKM.gk.ent)
call	Matched call

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Ferraro M.B., Giordani P., 2013. A new fuzzy clustering algorithm with entropy regularization. Proceedings of the meeting on Classification and Data Analysis (CLADAG).

See Also

FKM.gkb.ent, Fclust, Fclust.index, print.fclust, summary.fclust, plot.fclust, unemployment

Examples

```
## unemployment data
data(unemployment)
## Gustafson and Kessel-like fuzzy k-means with entropy regularization,
##fixing the number of clusters
clust=FKM.gk.ent(unemployment,k=3,ent=0.2,RS=10,stand=1)
## Not run:
## Gustafson and Kessel-like fuzzy k-means with entropy regularization,
##selecting the number of clusters
clust=FKM.gk.ent(unemployment,k=2:6,ent=0.2,RS=10,stand=1)

## End(Not run)
```

FKM.gk.ent.noise — *Gustafson and Kessel - like fuzzy k-means with entropy regularization and noise cluster*

Description

Performs the Gustafson and Kessel - like fuzzy *k*-means clustering algorithm with entropy regularization and noise cluster.
Differently from fuzzy *k*-means, it is able to discover non-spherical clusters.
The entropy regularization allows us to avoid using the artificial fuzziness parameter *m*. This is replaced by the degree of fuzzy entropy *ent*, related to the concept of temperature in statistical physics. An interesting property of the fuzzy *k*-means with entropy regularization is that the prototypes are obtained as weighted means with weights equal to the membership degrees (rather than to the membership degrees at the power of *m* as is for the fuzzy k-means).
The noise cluster is an additional cluster (with respect to the *k* standard clusters) such that objects recognized to be outliers are assigned to it with high membership degrees.

Usage

```
FKM.gk.ent.noise (X,k,ent,vp,delta,RS,stand,startU,index,alpha,conv,maxit,seed)
```

Arguments

X	Matrix or data.frame
k	An integer value or vector specifying the number of clusters for which the `index` is to be calculated (default: 2:6)
ent	Degree of fuzzy entropy (default: 1)
vp	Volume parameter (default: rep(1,k))
delta	Noise distance (default: average Euclidean distance between objects and prototypes from `FKM.gk.ent` using the same values of `k` and `m`)

RS	Number of (random) starts (default: 1)
stand	Standardization: if stand=1, the clustering algorithm is run using standardized data (default: no standardization)
startU	Rational start for the membership degree matrix U (default: no rational start)
index	Cluster validity index to select the number of clusters: "PC" (partition coefficient), "PE" (partition entropy), "MPC" (modified partition coefficient), "SIL" (silhouette), "SIL.F" (fuzzy silhouette), "XB" (Xie and Beni) (default: "SIL.F")
alpha	Weighting coefficient for the fuzzy silhouette index SIL.F (default: 1)
conv	Convergence criterion (default: 1e-9)
maxit	Maximum number of iterations (default: 1e+6)
seed	Seed value for random number generation (default: NULL)

Details

If startU is given, the argument k is ignored (the number of clusters is ncol(startU)).
If startU is given, the first element of value, cput and iter refer to the rational start.
If a cluster covariance matrix becomes singular, the algorithm stops and the element of value is NaN.
The default value for ent is in general not reasonable if FKM.gk.ent is run using raw data.
The update of the membership degrees requires the computation of exponential functions. In some cases, this may produce NaN values and the algorithm stops. Such a problem is usually solved by running FKM.gk.ent.noise using standardized data (stand=1).
The Babuska et al. variant in FKM.gkb.ent.noise is recommended.

Value

Object of class fclust, which is a list with the following components:

U	Membership degree matrix
H	Prototype matrix
F	Array containing the covariance matrices of all the clusters
clus	Matrix containing the indexes of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
medoid	Vector containing the indexes of the medoid objects (NULL for FKM.gk.ent.noise)
value	Vector containing the loss function values for the RS starts
criterion	Vector containing the values of the cluster validity index
iter	Vector containing the numbers of iterations for the RS starts
k	Number of clusters
m	Parameter of fuzziness (NULL for FKM.gk.ent.noise)
ent	Degree of fuzzy entropy
b	Parameter of the polynomial fuzzifier (NULL for FKM.gk.ent.noise)
vp	Volume parameter (default: rep(1,max(k)). If k is a vector, for each group the first k element of vpare considered.

`delta`	Noise distance
`gam`	Weighting parameter for the fuzzy covariance matrices (`NULL` for `FKM.ent.noise`)
`mcn`	Maximum condition number for the fuzzy covariance matrices (`NULL` for `FKM.ent.noise`)
`stand`	Standardization (Yes if `stand=1`, No if `stand=0`)
`Xca`	Data used in the clustering algorithm (standardized data if `stand=1`)
`X`	Raw data
`D`	Dissimilarity matrix (`NULL` for `FKM.ent.noise`)
`call`	Matched call

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Dave' R.N., 1991. Characterization and detection of noise in clustering. Pattern Recognition Letters, 12, 657-664.
Ferraro M.B., Giordani P., 2013. A new fuzzy clustering algorithm with entropy regularization. Proceedings of the meeting on Classification and Data Analysis (CLADAG).

See Also

`FKM.gkb.ent.noise`, `Fclust`, `Fclust.index`, `print.fclust`, `summary.fclust`, `plot.fclust`, `unemployment`

Examples

```
## Not run:
## unemployment data
data(unemployment)
## Gustafson and Kessel-like fuzzy k-means with entropy regularization and noise cluster,
##fixing the number of clusters
clust=FKM.gk.ent.noise(unemployment,k=3,ent=0.2,delta=1,RS=10,stand=1)
## Gustafson and Kessel-like fuzzy k-means with entropy regularization and noise cluster,
##selecting the number of clusters
clust=FKM.gk.ent.noise(unemployment,k=2:6,ent=0.2,delta=1,RS=10,stand=1)

## End(Not run)
```

FKM.gk.noise *Gustafson and Kessel - like fuzzy k-means with noise cluster*

Description

Performs the Gustafson and Kessel - like fuzzy *k*-means clustering algorithm with noise cluster.
Differently from fuzzy *k*-means, it is able to discover non-spherical clusters.
The noise cluster is an additional cluster (with respect to the *k* standard clusters) such that objects recognized to be outliers are assigned to it with high membership degrees.

Usage

```
FKM.gk.noise (X, k, m, vp, delta, RS, stand, startU, index, alpha, conv, maxit, seed)
```

Arguments

`X`	Matrix or data.frame
`k`	An integer value or vector specifying the number of clusters for which the `index` is to be calculated (default: 2:6)
`m`	Parameter of fuzziness (default: 2)
`vp`	Volume parameter (default: `rep(1,max(k))`). If `k` is a vector, for each group the first `k` element of `vp`are considered.
`delta`	Noise distance (default: average Euclidean distance between objects and prototypes from `FKM.gk` using the same values of `k` and `m`)
`RS`	Number of (random) starts (default: 1)
`stand`	Standardization: if `stand=1`, the clustering algorithm is run using standardized data (default: no standardization)
`startU`	Rational start for the membership degree matrix `U` (default: no rational start)
`index`	Cluster validity index to select the number of clusters: `"PC"` (partition coefficient), `"PE"` (partition entropy), `"MPC"` (modified partition coefficient), `"SIL"` (silhouette), `"SIL.F"` (fuzzy silhouette), `"XB"` (Xie and Beni) (default: "SIL.F")
`alpha`	Weighting coefficient for the fuzzy silhouette index `SIL.F` (default: 1)
`conv`	Convergence criterion (default: 1e-9)
`maxit`	Maximum number of iterations (default: 1e+6)
`seed`	Seed value for random number generation (default: NULL)

Details

If `startU` is given, the argument `k` is ignored (the number of clusters is `ncol(startU)`).
If `startU` is given, the first element of `value`, `cput` and `iter` refer to the rational start.
If a cluster covariance matrix becomes singular, then the algorithm stops and the element of `value` is NaN.
The Babuska et al. variant in `FKM.gkb.noise` is recommended.

Value

Object of class fclust, which is a list with the following components:

U	Membership degree matrix
H	Prototype matrix
F	Array containing the covariance matrices of all the clusters
clus	Matrix containing the indexes of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
medoid	Vector containing the indexes of the medoid objects (NULL for FKM.gk.noise)
value	Vector containing the loss function values for the RS starts
criterion	Vector containing the values of the cluster validity index
iter	Vector containing the numbers of iterations for the RS starts
k	Number of clusters
m	Parameter of fuzziness
ent	Degree of fuzzy entropy (NULL for FKM.gk.noise)
b	Parameter of the polynomial fuzzifier (NULL for FKM.gk.noise)
vp	Volume parameter
delta	Noise distance
gam	Weighting parameter for the fuzzy covariance matrices (NULL for FKM.gk.noise)
mcn	Maximum condition number for the fuzzy covariance matrices (NULL for FKM.gk.noise)
stand	Standardization (Yes if stand=1, No if stand=0)
Xca	Data used in the clustering algorithm (standardized data if stand=1)
X	Raw data
D	Dissimilarity matrix (NULL for FKM.gk.noise)
call	Matched call

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Dave' R.N., 1991. Characterization and detection of noise in clustering. Pattern Recognition Letters, 12, 657-664.
Gustafson E.E., Kessel W.C., 1978. Fuzzy clustering with a fuzzy covariance matrix. Proceedings of the IEEE Conference on Decision and Control, pp. 761-766.

See Also

FKM.gkb.noise, Fclust, Fclust.index, print.fclust, summary.fclust, plot.fclust, unemployment

Examples

```
## Not run:
## unemployment data
data(unemployment)
## Gustafson and Kessel-like fuzzy k-means with noise cluster, fixing the number of clusters
clust=FKM.gk.noise(unemployment,k=3,delta=20,RS=10)
## Gustafson and Kessel-like fuzzy k-means with noise cluster, selecting the number of clusters
clust=FKM.gk.noise(unemployment,k=2:6,delta=20,RS=10)

## End(Not run)
```

FKM.gkb *Gustafson, Kessel and Babuska - like fuzzy k-means*

Description

Performs the Gustafson, Kessel and Babuska - like fuzzy *k*-means clustering algorithm.
Differently from fuzzy *k*-means, it is able to discover non-spherical clusters.
The Babuska et al. variant improves the computation of the fuzzy covariance matrices in the standard Gustafson and Kessel clustering algorithm.

Usage

```
FKM.gkb (X, k, m, vp, gam, mcn, RS, stand, startU, index, alpha, conv, maxit, seed)
```

Arguments

`X`	Matrix or data.frame
`k`	An integer value or vector specifying the number of clusters for which the `index` is to be calculated (default: 2:6)
`m`	Parameter of fuzziness (default: 2)
`vp`	Volume parameter (default: rep(1,k))
`gam`	Weighting parameter for the fuzzy covariance matrices (default: 0)
`mcn`	Maximum condition number for the fuzzy covariance matrices (default: 1e+15)
`RS`	Number of (random) starts (default: 1)
`stand`	Standardization: if `stand=1`, the clustering algorithm is run using standardized data (default: no standardization)
`startU`	Rational start for the membership degree matrix `U` (default: no rational start)
`index`	Cluster validity index to select the number of clusters: `PC` (partition coefficient), `PE` (partition entropy), `MPC` (modified partition coefficient), `SIL` (silhouette), `SIL.F` (fuzzy silhouette), `XB` (Xie and Beni) (default: "SIL.F")
`alpha`	Weighting coefficient for the fuzzy silhouette index `SIL.F` (default: 1)
`conv`	Convergence criterion (default: 1e-9)
`maxit`	Maximum number of iterations (default: 1e+2)
`seed`	Seed value for random number generation (default: NULL)

Details

If `startU` is given, the argument `k` is ignored (the number of clusters is `ncol(startU)`).
If `startU` is given, the first element of `value`, `cput` and `iter` refer to the rational start.
If a cluster covariance matrix becomes singular, then the algorithm stops and the element of `value` is NaN.

Value

Object of class `fclust`, which is a list with the following components:

`U`	Membership degree matrix
`H`	Prototype matrix
`F`	Array containing the covariance matrices of all the clusters
`clus`	Matrix containing the indexes of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
`medoid`	Vector containing the indexes of the medoid objects (`NULL` for `FKM.gkb`)
`value`	Vector containing the loss function values for the `RS` starts
`criterion`	Vector containing the values of clustering index
`iter`	Vector containing the numbers of iterations for the `RS` starts
`k`	Number of clusters
`m`	Parameter of fuzziness
`ent`	Degree of fuzzy entropy (`NULL` for `FKM.gkb`)
`b`	Parameter of the polynomial fuzzifier (`NULL` for `FKM.gkb`)
`vp`	Volume parameter
`delta`	Noise distance (`NULL` for `FKM.gkb`)
`gam`	Weighting parameter for the fuzzy covariance matrices
`mcn`	Maximum condition number for the fuzzy covariance matrices
`stand`	Standardization (Yes if `stand=1`, No if `stand=0`)
`Xca`	Data used in the clustering algorithm (standardized data if `stand=1`)
`X`	Raw data
`D`	Dissimilarity matrix (`NULL` for `FKM.gkb`)
`call`	Matched call

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Babuska R., van der Veen P.J., Kaymak U., 2002. Improved covariance estimation for Gustafson-Kessel clustering. Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 1081-1085.
Gustafson E.E., Kessel W.C., 1978. Fuzzy clustering with a fuzzy covariance matrix. Proceedings of the IEEE Conference on Decision and Control, pp. 761-766.

See Also

FKM.gk, Fclust, Fclust.index, print.fclust, summary.fclust, plot.fclust, unemployment

Examples

```
## Not run:
## unemployment data
data(unemployment)
## Gustafson, Kessel and Babuska-like fuzzy k-means, fixing the number of clusters
clust=FKM.gkb(unemployment,k=3,RS=10)
## Gustafson, Kessel and Babuska-like fuzzy k-means, selecting the number of clusters
clust=FKM.gkb(unemployment,k=2:6,RS=10)
## End(Not run)
```

FKM.gkb.ent	*Gustafson, Kessel and Babuska - like fuzzy k-means with entropy regularization*

Description

Performs the Gustafson, Kessel and Babuska - like fuzzy k-means clustering algorithm with entropy regularization.
Differently from fuzzy k-means, it is able to discover non-spherical clusters.
The Babuska et al. variant improves the computation of the fuzzy covariance matrices in the standard Gustafson and Kessel clustering algorithm.
The entropy regularization allows us to avoid using the artificial fuzziness parameter m. This is replaced by the degree of fuzzy entropy *ent*, related to the concept of temperature in statistical physics. An interesting property of the fuzzy k-means with entropy regularization is that the prototypes are obtained as weighted means with weights equal to the membership degrees (rather than to the membership degrees at the power of m as is for the fuzzy k-means).

Usage

```
FKM.gkb.ent (X, k, ent, vp, gam, mcn, RS, stand, startU, index, alpha, conv, maxit, seed)
```

Arguments

X	Matrix or data.frame
k	An integer value or vector specifying the number of clusters for which the index is to be calculated (default: 2:6)
ent	Degree of fuzzy entropy (default: 1)
vp	Volume parameter (default: rep(1,k))
gam	Weighting parameter for the fuzzy covariance matrices (default: 0)
mcn	Maximum condition number for the fuzzy covariance matrices (default: 1e+15)
RS	Number of (random) starts (default: 1)

stand	Standardization: if stand=1, the clustering algorithm is run using standardized data (default: no standardization)
startU	Rational start for the membership degree matrix U (default: no rational start)
index	Cluster validity index to select the number of clusters: "PC" (partition coefficient), "PE" (partition entropy), "MPC" (modified partition coefficient), "SIL" (silhouette), "SIL.F" (fuzzy silhouette), "XB" (Xie and Beni) (default: "SIL.F")
alpha	Weighting coefficient for the fuzzy silhouette index SIL.F (default: 1)
conv	Convergence criterion (default: 1e-9)
maxit	Maximum number of iterations (default: 1e+2)
seed	Seed value for random number generation (default: NULL)

Details

If startU is given, the argument k is ignored (the number of clusters is ncol(startU)).
If startU is given, the first element of value, cput and iter refer to the rational start.
If a cluster covariance matrix becomes singular, the algorithm stops and the element of value is NaN.
The default value for ent is in general not reasonable if FKM.gk.ent is run using raw data.
The update of the membership degrees requires the computation of exponential functions. In some cases, this may produce NaN values and the algorithm stops. Such a problem is usually solved by running FKM.gk.ent using standardized data (stand=1).

Value

Object of class fclust, which is a list with the following components:

U	Membership degree matrix
H	Prototype matrix
F	Array containing the covariance matrices of all the clusters
clus	Matrix containing the indexes of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
medoid	Vector containing the indexes of the medoid objects (NULL for FKM.gkb.ent)
value	Vector containing the loss function values for the RS starts
criterion	Vector containing the values of the cluster validity index
iter	Vector containing the numbers of iterations for the RS starts
k	A integer value or vector indicating the number of clusters. (default: 2:6)
m	Parameter of fuzziness (NULL for FKM.gkb.ent)
ent	Degree of fuzzy entropy
b	Parameter of the polynomial fuzzifier (NULL for FKM.gkb.ent)
vp	Volume parameter (default: rep(1,max(k)). If k is a vector, for each group the first k element of vpare considered.
delta	Noise distance (NULL for FKM.gkb.ent)
gam	Weighting parameter for the fuzzy covariance matrices

mcn	Maximum condition number for the fuzzy covariance matrices
stand	Standardization (Yes if stand=1, No if stand=0)
Xca	Data used in the clustering algorithm (standardized data if stand=1)
X	Raw data
D	Dissimilarity matrix (NULL for FKM.gkb.ent)
call	Matched call

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Babuska R., van der Veen P.J., Kaymak U., 2002. Improved covariance estimation for Gustafson-Kessel clustering. Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 1081-1085.
Ferraro M.B., Giordani P., 2013. A new fuzzy clustering algorithm with entropy regularization. Proceedings of the meeting on Classification and Data Analysis (CLADAG).

See Also

FKM.gk.ent, Fclust, Fclust.index, print.fclust, summary.fclust, plot.fclust, unemployment

Examples

```
## Not run:
## unemployment data
data(unemployment)
## Gustafson, Kessel and Babuska-like fuzzy k-means with entropy regularization,
##fixing the number of clusters
clust=FKM.gkb.ent(unemployment,k=3,ent=0.2,RS=10,stand=1)
## Gustafson, Kessel and Babuska-like fuzzy k-means with entropy regularization,
##selecting the number of clusters
clust=FKM.gkb.ent(unemployment,k=2:6,ent=0.2,RS=10,stand=1)
## End(Not run)
```

FKM.gkb.ent.noise *Gustafson, Kessel and Babuska - like fuzzy k-means with entropy regularization and noise cluster*

Description

Performs the Gustafson, Kessel and Babuska - like fuzzy k-means clustering algorithm with entropy regularization and noise cluster.
Differently from fuzzy k-means, it is able to discover non-spherical clusters.
The Babuska et al. variant improves the computation of the fuzzy covariance matrices in the standard Gustafson and Kessel clustering algorithm.

The entropy regularization allows us to avoid using the artificial fuzziness parameter *m*. This is replaced by the degree of fuzzy entropy *ent*, related to the concept of temperature in statistical physics. An interesting property of the fuzzy *k*-means with entropy regularization is that the prototypes are obtained as weighted means with weights equal to the membership degrees (rather than to the membership degrees at the power of *m* as is for the fuzzy k-means).
The noise cluster is an additional cluster (with respect to the *k* standard clusters) such that objects recognized to be outliers are assigned to it with high membership degrees.

Usage

```
FKM.gkb.ent.noise (X,k,ent,vp,delta,gam,mcn,RS,stand,startU,index,alpha,conv,maxit,seed)
```

Arguments

`X`	Matrix or data.frame
`k`	An integer value or vector specifying the number of clusters for which the `index` is to be calculated (default: 2:6)
`ent`	Degree of fuzzy entropy (default: 1)
`vp`	Volume parameter (default: `rep(1,max(k))`). If `k` is a vector, for each group the first `k` element of `vp`are considered.
`delta`	Noise distance (default: average Euclidean distance between objects and prototypes from `FKM.gk.ent` using the same values of `k` and `m`)
`gam`	Weighting parameter for the fuzzy covariance matrices (default: 0)
`mcn`	Maximum condition number for the fuzzy covariance matrices (default: 1e+15)
`RS`	Number of (random) starts (default: 1)
`stand`	Standardization: if `stand=1`, the clustering algorithm is run using standardized data (default: no standardization)
`startU`	Rational start for the membership degree matrix `U` (default: no rational start)
`index`	Cluster validity index to select the number of clusters: `"PC"` (partition coefficient), `"PE"` (partition entropy), `"MPC"` (modified partition coefficient), `"SIL"` (silhouette), `"SIL.F"` (fuzzy silhouette), `"XB"` (Xie and Beni) (default: "SIL.F")
`alpha`	Weighting coefficient for the fuzzy silhouette index `SIL.F` (default: 1)
`conv`	Convergence criterion (default: 1e-9)
`maxit`	Maximum number of iterations (default: 1e+2)
`seed`	Seed value for random number generation (default: NULL)

Details

If `startU` is given, the argument `k` is ignored (the number of clusters is `ncol(startU)`).
If `startU` is given, the first element of `value`, `cput` and `iter` refer to the rational start.
If a cluster covariance matrix becomes singular, the algorithm stops and the element of `value` is NaN.
The default value for `ent` is in general not reasonable if `FKM.gk.ent` is run using raw data.
The update of the membership degrees requires the computation of exponential functions. In some cases, this may produce `NaN` values and the algorithm stops. Such a problem is usually solved by running `FKM.gk.ent.noise` using standardized data (`stand=1`).

Value

Object of class `fclust`, which is a list with the following components:

`U`	Membership degree matrix
`H`	Prototype matrix
`F`	Array containing the covariance matrices of all the clusters
`clus`	Matrix containing the indexes of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
`medoid`	Vector containing the indexes of the medoid objects (`NULL` for `FKM.gkb.ent.noise`)
`value`	Vector containing the loss function values for the `RS` starts
`criterion`	Vector containing the values of the cluster validity index
`iter`	Vector containing the numbers of iterations for the `RS` starts
`k`	Number of clusters
`m`	Parameter of fuzziness (`NULL` for `FKM.gkb.ent.noise`)
`ent`	Degree of fuzzy entropy
`b`	Parameter of the polynomial fuzzifier (`NULL` for `FKM.gkb.ent.noise`)
`vp`	Volume parameter
`delta`	Noise distance
`gam`	Weighting parameter for the fuzzy covariance matrices
`mcn`	Maximum condition number for the fuzzy covariance matrices
`stand`	Standardization (Yes if `stand=1`, No if `stand=0`)
`Xca`	Data used in the clustering algorithm (standardized data if `stand=1`)
`X`	Raw data
`D`	Dissimilarity matrix (`NULL` for `FKM.gkb.ent.noise`)
`call`	Matched call

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Babuska R., van der Veen P.J., Kaymak U., 2002. Improved covariance estimation for Gustafson-Kessel clustering. Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 1081-1085.
Dave' R.N., 1991. Characterization and detection of noise in clustering. Pattern Recognition Letters, 12, 657-664.
Ferraro M.B., Giordani P., 2013. A new fuzzy clustering algorithm with entropy regularization. Proceedings of the meeting on Classification and Data Analysis (CLADAG).

See Also

`FKM.gk.ent.noise`, `Fclust`, `Fclust.index`, `print.fclust`, `summary.fclust`, `plot.fclust`, `unemployment`

Examples

```
## Not run:
## unemployment data
data(unemployment)
## Gustafson, Kessel and Babuska-like fuzzy k-means with entropy regularization and noise cluster,
##fixing the number of clusters
clust=FKM.gkb.ent.noise(unemployment,k=3,ent=0.2,delta=1,RS=10,stand=1)
## Gustafson, Kessel and Babuska-like fuzzy k-means with entropy regularization and noise cluster,
##selecting the number of clusters
clust=FKM.gkb.ent.noise(unemployment,k=2:6,ent=0.2,delta=1,RS=10,stand=1)

## End(Not run)
```

`FKM.gkb.noise` *Gustafson, Kessel and Babuska - like fuzzy k-means with noise cluster*

Description

Performs the Gustafson, Kessel and Babuska - like fuzzy *k*-means clustering algorithm with noise cluster.
Differently from fuzzy *k*-means, it is able to discover non-spherical clusters.
The Babuska et al. variant improves the computation of the fuzzy covariance matrices in the standard Gustafson and Kessel clustering algorithm.
The noise cluster is an additional cluster (with respect to the *k* standard clusters) such that objects recognized to be outliers are assigned to it with high membership degrees.

Usage

```
FKM.gkb.noise (X,k,m,vp,delta,gam,mcn,RS,stand,startU,index,alpha,conv,maxit,seed)
```

Arguments

`X`	Matrix or data.frame
`k`	An integer value or vector specifying the number of clusters for which the `index` is to be calculated (default: 2:6)
`m`	Parameter of fuzziness (default: 2)
`vp`	Volume parameter (default: rep(1,k))
`delta`	Noise distance (default: average Euclidean distance between objects and prototypes from `FKM.gk` using the same values of `k` and `m`)
`gam`	Weighting parameter for the fuzzy covariance matrices (default: 0)
`mcn`	Maximum condition number for the fuzzy covariance matrices (default: 1e+15)
`RS`	Number of (random) starts (default: 1)
`stand`	Standardization: if `stand=1`, the clustering algorithm is run using standardized data (default: no standardization)
`startU`	Rational start for the membership degree matrix `U` (default: no rational start)

`index`	Cluster validity index to select the number of clusters: `"PC"` (partition coefficient), `"PE"` (partition entropy), `"MPC"` (modified partition coefficient), `"SIL"` (silhouette), `"SIL.F"` (fuzzy silhouette), `"XB"` (Xie and Beni) (default: "SIL.F")
`alpha`	Weighting coefficient for the fuzzy silhouette index `SIL.F` (default: 1)
`conv`	Convergence criterion (default: 1e-9)
`maxit`	Maximum number of iterations (default: 1e+2)
`seed`	Seed value for random number generation (default: NULL)

Details

If `startU` is given, the argument `k` is ignored (the number of clusters is `ncol(startU)`).
If `startU` is given, the first element of `value`, `cput` and `iter` refer to the rational start.
If a cluster covariance matrix becomes singular, then the algorithm stops and the element of `value` is NaN.

Value

Object of class `fclust`, which is a list with the following components:

`U`	Membership degree matrix
`H`	Prototype matrix
`F`	Array containing the covariance matrices of all the clusters
`clus`	Matrix containing the indexes of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
`medoid`	Vector containing the indexes of the medoid objects (`NULL` for `FKM.gkb.noise`)
`value`	Vector containing the loss function values for the `RS` starts
`criterion`	Vector containing the values of the cluster validity index
`iter`	Vector containing the numbers of iterations for the `RS` starts
`k`	Number of clusters
`m`	Parameter of fuzziness
`ent`	Degree of fuzzy entropy (`NULL` for `FKM.gkb.noise`)
`b`	Parameter of the polynomial fuzzifier (`NULL` for `FKM.gkb.noise`)
`vp`	Volume parameter (default: `rep(1,max(k))`). If `k` is a vector, for each group the first `k` element of vpare considered.
`delta`	Noise distance
`gam`	Weighting parameter for the fuzzy covariance matrices
`mcn`	Maximum condition number for the fuzzy covariance matrices
`stand`	Standardization (Yes if `stand=1`, No if `stand=0`)
`Xca`	Data used in the clustering algorithm (standardized data if `stand=1`)
`X`	Raw data
`D`	Dissimilarity matrix (`NULL` for `FKM.gkb.noise`)
`call`	Matched call

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Babuska R., van der Veen P.J., Kaymak U., 2002. Improved covariance estimation for Gustafson-Kessel clustering. Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 1081-1085.
Dave' R.N., 1991. Characterization and detection of noise in clustering. Pattern Recognition Letters, 12, 657-664.
Gustafson E.E., Kessel W.C., 1978. Fuzzy clustering with a fuzzy covariance matrix. Proceedings of the IEEE Conference on Decision and Control, pp. 761-766.

See Also

`FKM.gk.noise`, `Fclust`, `Fclust.index`, `print.fclust`, `summary.fclust`, `plot.fclust`, `unemployment`

Examples

```
## Not run:
## unemployment data
data(unemployment)
## Gustafson, Kessel and Babuska-like fuzzy k-means with noise cluster,
##fixing the number of clusters
clust=FKM.gkb.noise(unemployment,k=3,delta=20,RS=10)
## Gustafson, Kessel and Babuska-like fuzzy k-means with noise cluster,
##selecting the number of clusters
clust=FKM.gkb.noise(unemployment,k=2:6,delta=20,RS=10)
## End(Not run)
```

`FKM.med` *Fuzzy k-medoids*

Description

Performs the fuzzy k-medoids clustering algorithm.
Differently from fuzzy k-means where the cluster prototypes (centroids) are artificial objects computed as weighted means, in the fuzzy k-medoids the cluster prototypes (medoids) are a subset of the observed objects.

Usage

```
FKM.med (X, k, m, RS, stand, startU, index, alpha, conv, maxit, seed)
```

Arguments

X	Matrix or data.frame
k	An integer value or vector indicating the number of clusters (default: 2:6)
m	Parameter of fuzziness (default: 1.5)
RS	Number of (random) starts (default: 1)
stand	Standardization: if stand=1, the clustering algorithm is run using standardized data (default: no standardization)
startU	Rational start for the membership degree matrix U (default: no rational start)
index	Cluster validity index to select the number of clusters: PC (partition coefficient), PE (partition entropy), MPC (modified partition coefficient), SIL (silhouette), SIL.F (fuzzy silhouette), XB (Xie and Beni) (default: "SIL.F")
alpha	Weighting coefficient for the fuzzy silhouette index SIL.F (default: 1)
conv	Convergence criterion (default: 1e-9)
maxit	Maximum number of iterations (default: 1e+6)
seed	Seed value for random number generation (default: NULL)

Details

If startU is given, the argument k is ignored (the number of clusters is ncol(startU)).
If startU is given, the first element of value, cput and iter refer to the rational start.
In FKM.med the parameter of fuzziness is usually lower than the one used in FKM.

Value

Object of class fclust, which is a list with the following components:

U	Membership degree matrix
H	Prototype matrix
F	Array containing the covariance matrices of all the clusters (NULL for FKM.med)
clus	Matrix containing the indexes of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
medoid	Vector containing the indexes of the medoid objects
value	Vector containing the loss function values for the RS starts
criterion	Vector containing the values of the cluster validity index
iter	Vector containing the numbers of iterations for the RS starts
k	Number of clusters
m	Parameter of fuzziness
ent	Degree of fuzzy entropy (NULL for FKM.med)
b	Parameter of the polynomial fuzzifier (NULL for FKM.med)
vp	Volume parameter (NULL for FKM.med)
delta	Noise distance (NULL for FKM.med)

gam	Weighting parameter for the fuzzy covariance matrices (NULL for FKM.med)
mcn	Maximum condition number for the fuzzy covariance matrices (NULL for FKM.med)
stand	Standardization (Yes if stand=1, No if stand=0)
Xca	Data used in the clustering algorithm (standardized data if stand=1)
X	Raw data
D	Dissimilarity matrix (NULL for FKM.med)
call	Matched call

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Krishnapuram R., Joshi A., Nasraoui O., Yi L., 2001. Low-complexity fuzzy relational clustering algorithms for web mining. IEEE Transactions on Fuzzy Systems, 9, 595-607.

See Also

FKM.med.noise, Fclust, Fclust.index, print.fclust, summary.fclust, plot.fclust, Mc

Examples

```
## Not run:
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
## fuzzy k-medoids, fixing the number of clusters
## (excluded the factor column Type (last column))
clust=FKM.med(Mc[,1:(ncol(Mc)-1)],k=6,m=1.1,RS=10,stand=1)
## fuzzy k-medoids, selecting the number of clusters
## (excluded the factor column Type (last column))
clust=FKM.med(Mc[,1:(ncol(Mc)-1)],k=2:6,m=1.1,RS=10,stand=1)

## End(Not run)
```

FKM.med.noise *Fuzzy k-medoids with noise cluster*

Description

Performs the fuzzy k-medoids clustering algorithm with noise cluster.
Differently from fuzzy k-means where the cluster prototypes (centroids) are artificial objects computed as weighted means, in the fuzzy k-medoids the cluster prototypes (medoids) are a subset of the observed objects.
The noise cluster is an additional cluster (with respect to the k standard clusters) such that objects recognized to be outliers are assigned to it with high membership degrees.

Usage

```
FKM.med.noise (X, k, m, delta, RS, stand, startU, index, alpha, conv, maxit, seed)
```

Arguments

X	Matrix or data.frame
k	An integer value or vector specifying the number of clusters for which the index is to be calculated (default: 2:6)
m	Parameter of fuzziness (default: 1.5)
delta	Noise distance (default: average Euclidean distance between objects and prototypes from FKM.med using the same values of k and m)
RS	Number of (random) starts (default: 1)
stand	Standardization: if stand=1, the clustering algorithm is run using standardized data (default: no standardization)
startU	Rational start for the membership degree matrix U (default: no rational start)
index	Cluster validity index to select the number of clusters: PC (partition coefficient), PE (partition entropy), MPC (modified partition coefficient), SIL (silhouette), SIL.F (fuzzy silhouette), XB (Xie and Beni) (default: "SIL.F")
alpha	Weighting coefficient for the fuzzy silhouette index SIL.F (default: 1)
conv	Convergence criterion (default: 1e-9)
maxit	Maximum number of iterations (default: 1e+6)
seed	Seed value for random number generation (default: NULL)

Details

If startU is given, the argument k is ignored (the number of clusters is ncol(startU)).
If startU is given, the first element of value, cput and iter refer to the rational start.
As for FKM.med, in FKM.med.noise the parameter of fuzziness is usually lower than the one used in FKM.

Value

Object of class `fclust`, which is a list with the following components:

`U`	Membership degree matrix
`H`	Prototype matrix
`F`	Array containing the covariance matrices of all the clusters (`NULL` for `FKM.med.noise`)
`clus`	Matrix containing the indexes of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
`medoid`	Vector containing the indexes of the medoid objects
`value`	Vector containing the loss function values for the `RS` starts
`criterion`	Vector containing the values of clustering index
`iter`	Vector containing the numbers of iterations for the `RS` starts
`k`	Number of clusters
`m`	Parameter of fuzziness
`ent`	Degree of fuzzy entropy (`NULL` for `FKM.med.noise`)
`b`	Parameter of the polynomial fuzzifier (`NULL` for `FKM.med.noise`)
`vp`	Volume parameter (`NULL` for `FKM.med.noise`)
`delta`	Noise distance
`gam`	Weighting parameter for the fuzzy covariance matrices (`NULL` for `FKM.med.noise`)
`mcn`	Maximum condition number for the fuzzy covariance matrices (`NULL` for `FKM.med.noise`)
`stand`	Standardization (Yes if `stand=1`, No if `stand=0`)
`Xca`	Data used in the clustering algorithm (standardized data if `stand=1`)
`X`	Raw data
`D`	Dissimilarity matrix (`NULL` for `FKM.med.noise`)
`call`	Matched call

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Dave' R.N., 1991. Characterization and detection of noise in clustering. Pattern Recognition Letters, 12, 657-664.
Krishnapuram R., Joshi A., Nasraoui O., Yi L., 2001. Low-complexity fuzzy relational clustering algorithms for web mining. IEEE Transactions on Fuzzy Systems, 9, 595-607.

See Also

`FKM.med`, `Fclust`, `Fclust.index`, `print.fclust`, `summary.fclust`, `plot.fclust`, `butterfly`

Examples

```
## butterfly data
data(butterfly)
## fuzzy k-medoids with noise cluster, fixing the number of clusters
clust=FKM.med.noise(butterfly,k=2,RS=5,delta=3)
## fuzzy k-medoids with noise cluster, selecting the number of clusters
clust=FKM.med.noise(butterfly,RS=5,delta=3)
```

`FKM.noise` *Fuzzy k-means with noise cluster*

Description

Performs the fuzzy k-means clustering algorithm with noise cluster.
The noise cluster is an additional cluster (with respect to the k standard clusters) such that objects recognized to be outliers are assigned to it with high membership degrees.

Usage

```
FKM.noise (X, k, m, delta, RS, stand, startU, index, alpha, conv, maxit, seed)
```

Arguments

`X`	Matrix or data.frame
`k`	An integer value or vector specifying the number of clusters for which the `index` is to be calculated (default: 2:6)
`m`	Parameter of fuzziness (default: 2)
`delta`	Noise distance (default: average Euclidean distance between objects and prototypes from `FKM` using the same values of `k` and `m`)
`RS`	Number of (random) starts (default: 1)
`stand`	Standardization: if `stand=1`, the clustering algorithm is run using standardized data (default: no standardization)
`startU`	Rational start for the membership degree matrix `U` (default: no rational start)
`index`	Cluster validity index to select the number of clusters: `PC` (partition coefficient), `PE` (partition entropy), `MPC` (modified partition coefficient), `SIL` (silhouette), `SIL.F` (fuzzy silhouette), `XB` (Xie and Beni) (default: "SIL.F")
`alpha`	Weighting coefficient for the fuzzy silhouette index `SIL.F` (default: 1)
`conv`	Convergence criterion (default: 1e-9)
`maxit`	Maximum number of iterations (default: 1e+6)
`seed`	Seed value for random number generation (default: NULL)

Details

If `startU` is given, the argument `k` is ignored (the number of clusters is `ncol(startU)`).
If `startU` is given, the first element of `value`, `cput` and `iter` refer to the rational start.

Value

Object of class `fclust`, which is a list with the following components:

`U`	Membership degree matrix
`H`	Prototype matrix
`F`	Array containing the covariance matrices of all the clusters (`NULL` for `FKM.noise`)
`clus`	Matrix containing the indexes of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
`medoid`	Vector containing the indexes of the medoid objects (`NULL` for `FKM.noise`)
`value`	Vector containing the loss function values for the `RS` starts
`criterion`	Vector containing the values of the cluster validity index
`iter`	Vector containing the numbers of iterations for the `RS` starts
`k`	Number of clusters
`m`	Parameter of fuzziness
`ent`	Degree of fuzzy entropy (`NULL` for `FKM.noise`)
`b`	Parameter of the polynomial fuzzifier (`NULL` for `FKM.noise`)
`vp`	Volume parameter (`NULL` for `FKM.noise`)
`delta`	Noise distance
`gam`	Weighting parameter for the fuzzy covariance matrices (`NULL` for `FKM.noise`)
`mcn`	Maximum condition number for the fuzzy covariance matrices (`NULL` for `FKM.noise`)
`stand`	Standardization (Yes if `stand=1`, No if `stand=0`)
`Xca`	Data used in the clustering algorithm (standardized data if `stand=1`)
`X`	Raw data
`D`	Dissimilarity matrix (`NULL` for `FKM.noise`)
`call`	Matched call

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Dave' R.N., 1991. Characterization and detection of noise in clustering. Pattern Recognition Letters, 12, 657-664.

See Also

`FKM`, `Fclust`, `Fclust.index`, `print.fclust`, `summary.fclust`, `plot.fclust`, `butterfly`

Examples

```
## butterfly data
data(butterfly)
## fuzzy k-means with noise cluster, fixing the number of clusters
clust=FKM.noise(butterfly, k = 2, RS=5,delta=3)
## fuzzy k-means with noise cluster, selecting the number of clusters
clust=FKM.noise(butterfly,RS=5,delta=3)
```

`FKM.pf` *Fuzzy k-means with polynomial fuzzifier*

Description

Performs the fuzzy k-means clustering algorithm with polynomial fuzzifier function.
The polynomial fuzzifier creates areas of crisp membership degrees around the prototypes while, outside of these areas of crisp membership degrees, fuzzy membership degrees are given. Therefore, the polynomial fuzzifier produces membership degrees equal to one for objects clearly assigned to clusters, that is, very close to the cluster prototypes.

Usage

```
FKM.pf (X, k, b, RS, stand, startU, index, alpha, conv, maxit, seed)
```

Arguments

`X`	Matrix or data.frame
`k`	An integer value or vector specifying the number of clusters for which the `index` is to be calculated (default: 2:6)
`b`	Parameter of the polynomial fuzzifier (default: 0.5)
`RS`	Number of (random) starts (default: 1)
`stand`	Standardization: if `stand=1`, the clustering algorithm is run using standardized data (default: no standardization)
`startU`	Rational start for the membership degree matrix `U` (default: no rational start)
`index`	Cluster validity index to select the number of clusters: `"PC"` (partition coefficient), `"PE"` (partition entropy), `"MPC"` (modified partition coefficient), `"SIL"` (silhouette), `"SIL.F"` (fuzzy silhouette), `"XB"` (Xie and Beni) (default: "SIL.F")
`alpha`	Weighting coefficient for the fuzzy silhouette index `SIL.F` (default: 1)
`conv`	Convergence criterion (default: 1e-9)
`maxit`	Maximum number of iterations (default: 1e+6)
`seed`	Seed value for random number generation (default: NULL)

Details

If `startU` is given, the argument `k` is ignored (the number of clusters is `ncol(startU)`).
If `startU` is given, the first element of `value`, `cput` and `iter` refer to the rational start.

Value

Object of class `fclust`, which is a list with the following components:

`U`	Membership degree matrix
`H`	Prototype matrix
`F`	Array containing the covariance matrices of all the clusters (`NULL` for `FKM.pf`)
`clus`	Matrix containing the indexes of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
`medoid`	Vector containing the indexes of the medoid objects (`NULL` for `FKM.pf`)
`value`	Vector containing the loss function values for the `RS` starts
`criterion`	Vector containing the values of the cluster validity index
`iter`	Vector containing the numbers of iterations for the `RS` starts
`k`	Number of clusters
`m`	Parameter of fuzziness (`NULL` for `FKM.pf`)
`ent`	Degree of fuzzy entropy (`NULL` for `FKM.pf`)
`b`	Parameter of the polynomial fuzzifier
`vp`	Volume parameter (`NULL` for `FKM.pf`)
`delta`	Noise distance (`NULL` for `FKM.pf`)
`gam`	Weighting parameter for the fuzzy covariance matrices (`NULL` for `FKM.pf`)
`mcn`	Maximum condition number for the fuzzy covariance matrices (`NULL` for `FKM.pf`)
`stand`	Standardization (Yes if `stand=1`, No if `stand=0`)
`Xca`	Data used in the clustering algorithm (standardized data if `stand=1`)
`X`	Raw data
`D`	Dissimilarity matrix (`NULL` for `FKM.pf`)
`call`	Matched call

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Winkler R., Klawonn F., Hoeppner F., Kruse R., 2010. Fuzzy Cluster Analysis of Larger Data Sets. In: Scalable Fuzzy Algorithms for Data Management and Analysis: Methods and Design IGI Global, pp. 302-331. IGI Global, Hershey.
Winkler R., Klawonn F., Kruse R., 2011. Fuzzy clustering with polynomial fuzzifier function in connection with M-estimators. Applied and Computational Mathematics, 10, 146-163.

See Also

`FKM.pf.noise`, `Fclust`, `Fclust.index`, `print.fclust`, `summary.fclust`, `plot.fclust`, `Mc`

Examples

```
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
## fuzzy k-means with polynomial fuzzifier, fixing the number of clusters
## (excluded the factor column Type (last column))
clust=FKM.pf(Mc[,1:(ncol(Mc)-1)],k=6,stand=1)
## fuzzy k-means with polynomial fuzzifier, selecting the number of clusters
## (excluded the factor column Type (last column))
clust=FKM.pf(Mc[,1:(ncol(Mc)-1)],k=2:6,stand=1)
```

FKM.pf.noise — *Fuzzy k-means with polynomial fuzzifier and noise cluster*

Description

Performs the fuzzy *k*-means clustering algorithm with polynomial fuzzifier function and noise cluster.
The polynomial fuzzifier creates areas of crisp membership degrees around the prototypes while, outside of these areas of crisp membership degrees, fuzzy membership degrees are given. Therefore, the polynomial fuzzifier produces membership degrees equal to one for objects clearly assigned to clusters, that is, very close to the cluster prototypes.
The noise cluster is an additional cluster (with respect to the *k* standard clusters) such that objects recognized to be outliers are assigned to it with high membership degrees.

Usage

```
FKM.pf.noise (X, k, b, delta, RS, stand, startU, index, alpha, conv, maxit, seed)
```

Arguments

`X`	Matrix or data.frame
`k`	An integer value or vector specifying the number of clusters for which the `index` is to be calculated (default: 2:6)
`b`	Parameter of the polynomial fuzzifier (default: 0.5)
`delta`	Noise distance (default: average Euclidean distance between objects and prototypes from `FKM.pf` using the same values of `k` and `m`)
`RS`	Number of (random) starts (default: 1)
`stand`	Standardization: if `stand=1`, the clustering algorithm is run using standardized data (default: no standardization)
`startU`	Rational start for the membership degree matrix `U` (default: no rational start)

`index`	Cluster validity index to select the number of clusters: `PC` (partition coefficient), `PE` (partition entropy), `MPC` (modified partition coefficient), `SIL` (silhouette), `SIL.F` (fuzzy silhouette), `XB` (Xie and Beni) (default: "SIL.F")
`alpha`	Weighting coefficient for the fuzzy silhouette index `SIL.F` (default: 1)
`conv`	Convergence criterion (default: 1e-9)
`maxit`	Maximum number of iterations (default: 1e+6)
`seed`	Seed value for random number generation (default: NULL)

Details

If `startU` is given, the argument `k` is ignored (the number of clusters is `ncol(startU)`).
If `startU` is given, the first element of `value`, `cput` and `iter` refer to the rational start.

Value

Object of class `fclust`, which is a list with the following components:

`U`	Membership degree matrix
`H`	Prototype matrix
`F`	Array containing the covariance matrices of all the clusters (`NULL` for `FKM.pf.noise`)
`clus`	Matrix containing the indexes of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
`medoid`	Vector containing the indexes of the medoid objects (`NULL` for `FKM.pf.noise`)
`value`	Vector containing the loss function values for the `RS` starts
`criterion`	Vector containing the values of the cluster validity index
`iter`	Vector containing the numbers of iterations for the `RS` starts
`k`	Number of clusters
`m`	Parameter of fuzziness (`NULL` for `FKM.pf.noise`)
`ent`	Degree of fuzzy entropy (`NULL` for `FKM.pf.noise`)
`b`	Parameter of the polynomial fuzzifier
`vp`	Volume parameter (`NULL` for `FKM.pf.noise`)
`delta`	Noise distance
`gam`	Weighting parameter for the fuzzy covariance matrices (`NULL` for `FKM.pf.noise`)
`mcn`	Maximum condition number for the fuzzy covariance matrices (`NULL` for `FKM.pf.noise`)
`stand`	Standardization (Yes if `stand=1`, No if `stand=0`)
`Xca`	Data used in the clustering algorithm (standardized data if `stand=1`)
`X`	Raw data
`D`	Dissimilarity matrix (`NULL` for `FKM.pf.noise`)
`call`	Matched call

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Dave' R.N., 1991. Characterization and detection of noise in clustering. Pattern Recognition Letters, 12, 657-664.
Winkler R., Klawonn F., Hoeppner F., Kruse R., 2010. Fuzzy cluster analysis of larger data sets. In: Scalable Fuzzy Algorithms for Data Management and Analysis: Methods and Design IGI Global, pp. 302-331. IGI Global, Hershey.
Winkler R., Klawonn F., Kruse R., 2011. Fuzzy clustering with polynomial fuzzifier function in connection with M-estimators. Applied and Computational Mathematics, 10, 146-163.

See Also

FKM.pf, Fclust, Fclust.index, print.fclust, summary.fclust, plot.fclust, Mc

Examples

```
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
## fuzzy k-means with polynomial fuzzifier and noise cluster, fixing the number of clusters
## (excluded the factor column Type (last column))
clust=FKM.pf.noise(Mc[,1:(ncol(Mc)-1)],k=6,stand=1)
## fuzzy k-means with polynomial fuzzifier and noise cluster, selecting the number of clusters
## (excluded the factor column Type (last column))
clust=FKM.pf.noise(Mc[,1:(ncol(Mc)-1)],k=2:6,stand=1)
```

---

houseVotes *Congressional Voting Records Data*

Description

1984 United Stated Congressional Voting Records for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the Congressional Quarterly Almanac.

Usage

```
data(houseVotes)
```

Format

A data.frame with 435 rows on 17 columns (16 qualitative variables and 1 classification variable).

Details

The data collect 1984 United Stated Congressional Voting Records for each of the 435 U.S. House of Representatives Congressmen on the 16 key votes identified by the Congressional Quarterly Almanac (CQA). The variable `class` splits the observations in `democrat` and `republican`. The qualitative variables refer to the votes on `handicapped-infants`, `water-project-cost-sharing`, `adoption-of-the-budget-resolution`, `physician-fee-freeze`, `el-salvador-aid`, `religious-groups-in-schools`, `anti-satellite-test-ban`, `aid-to-nicaraguan-contras`, `mx-missile`, `immigration`, `synfuels-corporation-cutbac` `education-spending`, `superfund-right-to-sue`, `crime`, `duty-free-exports`, and `export-administration-act-south` All these 16 variables are objects of class `factor` with three levels according to the CQA scheme: `y` refers to the types of votes "voted for", "paired for" and "announced for"; `n` to "voted against", "paired against" and "announced against"; codeyn to "voted present", "voted present to avoid conflict of interest" and "did not vote or otherwise make a position known".

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

Source

https://archive.ics.uci.edu/ml/datasets/congressional+voting+records

References

Schlimmer, J.C., 1987. Concept acquisition through representational adjustment. Doctoral dissertation, Department of Information and Computer Science, University of California, Irvine, CA.

See Also

`NEFRC`, `NEFRC.noise`

Examples

```
data(houseVotes)
X=houseVotes[,-1]
class=houseVotes[,1]
```

`Hraw` *Raw prototypes*

Description

Produces prototypes using the original units of measurement of *X* (useful if the clustering algorithm is run using standardized data).

Usage

```
Hraw (X, H)
```

Arguments

X	Matrix or data.frame
H	Prototype matrix

Value

Hraw	Prototypes matrix using the original units of measurement of X

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

See Also

Fclust, unemployment

Examples

```
## example n.1 (k-means case)
## unemployment data
data(unemployment)
## fuzzy k-means
unempFKM=FKM(unemployment,k=3,stand=1)
## standardized prototypes
unempFKM$H
## prototypes using the original units of measurement
unempFKM$Hraw=Hraw(unempFKM$X,unempFKM$H)
## example n.2  (k-medoids case)
## unemployment data
data(unemployment)
## fuzzy k-medoids
## Not run:
## It may take more than a few seconds
unempFKM.med=FKM.med(unemployment,k=3,RS=10,stand=1)
## prototypes using the original units of measurement:
## in fuzzy k-medoids one can equivalently use
unempFKM.med$Hraw1=Hraw(unempFKM.med$X,unempFKM.med$H)
unempFKM.med$Hraw2=unempFKM.med$X[unempFKM.med$medoid,]
## End(Not run)
```

JACCARD.F *Fuzzy Jaccard index*

Description

Produces the fuzzy version of the Jaccard index between a hard (reference) partition and a fuzzy partition.

Usage

```
JACCARD.F(VC, U, t_norm)
```

Arguments

VC	Vector of class labels
U	Fuzzy membership degree matrix or data.frame
t_norm	Type of the triangular norm: "minimum" (minimum triangular norm), "triangular product" (product norm) (default: "minimum")

Value

jaccard.fValue of the fuzzy Jaccard index

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Campello, R.J., 2007. A fuzzy extension of the Rand index and other related indexes for clustering and classification assessment. Pattern Recognition Letters, 28, 833-841.
Jaccard, P., 1901. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin de la Société Vaudoise des Sciences Naturelles, 37, 547-579.

See Also

ARI.F, RI.F, Fclust.compare

Examples

```
## Not run:
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
## fuzzy k-means
## (excluded the factor column Type (last column))
clust=FKM(Mc[,1:(ncol(Mc)-1)],k=6,m=1.5,stand=1)
## fuzzy Jaccard index
jaccard.f=JACCARD.F(VC=Mc$Type,U=clust$U)

## End(Not run)
```

Mc *McDonald's data*

Description

Nutrition analysis of McDonald's menu items.

Usage

```
data(Mc)
```

Format

A data.frame with 81 rows and 16 columns.

Details

Data are from McDonald's USA Nutrition Facts for Popular Menu Items. A subset of menu items is reported. Beverages are excluded. In case of duplications, regular size or medium size information is reported. The variable Type is a factor the levels of which specify the kind of the menu items. Although some menu items could be well described by more than one level, only one level of the variable Type specifies each menu item. Percent Daily Values (%DV) are based on a 2,000 calorie diet. Some menu items are registered trademarks.

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

See Also

Fclust, FKM, FKM.ent, FKM.med

Examples

```
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
p=(ncol(Mc)-1)
## fuzzy k-means (excluded the factor column Type (last column))
clust.FKM=FKM(Mc[,1:p],k=6,m=1.5,stand=1)
## new factor column Cluster.FKM containing the cluster assignment information
## using fuzzy k-means
Mc[,ncol(Mc)+1]=factor(clust.FKM$clus[,1])
colnames(Mc)[ncol(Mc)]=("Cluster.FKM")
```

```
levels(Mc$Cluster.FKM)=paste("Clus FKM",1:clust.FKM$k,sep=" ")
## contingency table (Cluster.FKM vs Type)
## to assess whether clusters can be interpreted in terms of the levels of Type
table(Mc$Type,Mc$Cluster.FKM)
## prototypes using the original units of measurement
clust.FKM$Hraw=Hraw(clust.FKM$X,clust.FKM$H)
clust.FKM$Hraw
## fuzzy k-means with entropy regularization
## (excluded the factor column Type (last column))
## Not run:
## It may take more than a few seconds
clust.FKM.ent=FKM.ent(Mc[,1:p],k=6,ent=3,RS=10,stand=1)
## new factor column Cluster.FKM.ent containing the cluster assignment information
## using fuzzy k-medoids with entropy regularization
Mc[,ncol(Mc)+1]=factor(clust.FKM.ent$clus[,1])
colnames(Mc)[ncol(Mc)]=("Cluster.FKM.ent")
levels(Mc$Cluster.FKM.ent)=paste("Clus FKM.ent",1:clust.FKM.ent$k,sep=" ")
## contingency table (Cluster.FKM.ent vs Type)
## to assess whether clusters can be interpreted in terms of the levels of Type
table(Mc$Type,Mc$Cluster.FKM.ent)
## prototypes using the original units of measurement
clust.FKM.ent$Hraw=Hraw(clust.FKM.ent$X,clust.FKM.ent$H)
clust.FKM.ent$Hraw
## End(Not run)
## fuzzy k-medoids
## (excluded the factor column Type (last column))
clust.FKM.med=FKM.med(Mc[,1:p],k=6,m=1.1,RS=10,stand=1)
## new factor column Cluster.FKM.med containing the cluster assignment information
## using fuzzy k-medoids with entropy regularization
Mc[,ncol(Mc)+1]=factor(clust.FKM.med$clus[,1])
colnames(Mc)[ncol(Mc)]=("Cluster.FKM.med")
levels(Mc$Cluster.FKM.med)=paste("Clus FKM.med",1:clust.FKM.med$k,sep=" ")
## contingency table (Cluster.FKM.med vs Type)
## to assess whether clusters can be interpreted in terms of the levels of Type
table(Mc$Type,Mc$Cluster.FKM.med)
## prototypes using the original units of measurement
clust.FKM.med$Hraw=Hraw(clust.FKM.med$X,clust.FKM.med$H)
clust.FKM.med$Hraw
## or, equivalently,
Mc[clust.FKM.med$medoid,1:p]
```

MPC *Modified partition coefficient*

Description

Produces the modified partition coefficient index. The optimal number of clusters k is such that the index takes the maximum value.

Usage

```
MPC (U)
```

Arguments

U	Membership degree matrix

Value

mpc	Value of the modified partition coefficient index

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Dave' R.N., 1996. Validating fuzzy partitions obtained through *c*-shells clustering. Pattern Recognition Letters, 17, 613-623.

See Also

PC, PE, SIL, SIL.F, XB, Fclust, Mc

Examples

```
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
## fuzzy k-means
## (excluded the factor column Type (last column))
clust=FKM(Mc[,1:(ncol(Mc)-1)],k=6,m=1.5,stand=1)
## modified partition coefficient
mpc=MPC(clust$U)
```

NBA *NBA teams data*

Description

NBA team statistics from the 2017-2018 regular season.

Usage

```
data(NBA)
```

Format

A data.frame with 30 rows and 22 columns.

Details

Data refer to some statistics of the NBA teams for the regular season 2017-2018. The teams are distinguished according to two classification variables.

The statistics are: number of wins (`W`), field goals made (`FGM`), field goals attempted (`FGA`), field goals percentage (`FGP`), 3 point field goals made (`3PM`), 3 point field goals attempted (`3PA`), 3 point field goals percentage (`3PP`), free throws made (`FTM`), free throws attempted (`FTA`), free throws percentage (`FTP`), offensive rebounds (`OREB`), defensive rebounds (`DREB`), assists (`AST`), turnovers (`TOV`), steals (`STL`), blocks (`BLK`), blocked field goal attempts (`BLKA`), personal fouls (`PF`), personal fouls drawn (`PFD`) and points (`PTS`). Moreover, reported are the conference (`Conference`) and the playoff appearance (`Playoff`).

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

Source

https://stats.nba.com/teams/traditional/

See Also

`FKM`

Examples

```
## Not run:

data(NBA)
## A subset of variables is considered
X <- NBA[,c(4,7,10,11,12,13,14,15,16,17,20)]
clust.FKM=FKM(X=X,k=2:6,m=1.5,RS=50,stand=1,index="SIL.F",alpha=1)
summary(clust.FKM)

## End(Not run)
```

NEFRC *Non-Euclidean Fuzzy Relational Clustering*

Description

Performs the Non-Euclidean Fuzzy Relational data Clustering algorithm.

Usage

```
NEFRC(D, k, m, RS, startU, index, alpha, conv, maxit, seed)
```

Arguments

`D`	Matrix or data.frame containing distances/dissimilarities
`k`	An integer value or vector specifying the number of clusters for which the `index` is to be calculated (default: 2:6)
`m`	Parameter of fuzziness (default: 2)
`RS`	Number of (random) starts (default: 1)
`startU`	Rational start for the membership degree matrix `U` (default: no rational start)
`conv`	Convergence criterion (default: 1e-9)
`index`	Cluster validity index to select the number of clusters: `"PC"` (partition coefficient), `"PE"` (partition entropy), `"MPC"` (modified partition coefficient), `"SIL"` (silhouette), `"SIL.F"` (fuzzy silhouette) (default: "SIL.F")
`alpha`	Weighting coefficient for the fuzzy silhouette index `SIL.F` (default: 1)
`maxit`	Maximum number of iterations (default: 1e+6)
`seed`	Seed value for random number generation (default: NULL)

Details

If `startU` is given, the argument `k` is ignored (the number of clusters is `ncol(startU)`).
If `startU` is given, the first element of `value`, `cput` and `iter` refer to the rational start.

Value

Object of class `fclust`, which is a list with the following components:

`U`	Membership degree matrix
`H`	Prototype matrix (`NULL` for `NEFRC`)
`F`	Array containing the covariance matrices of all the clusters (`NULL` for `NEFRC`).
`clus`	Matrix containing the indexes of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
`medoid`	Vector containing the indexes of the medoid objects (`NULL` for `NEFRC`).
`value`	Vector containing the loss function values for the `RS` starts
`criterion`	Vector containing the values of the cluster validity index
`iter`	Vector containing the numbers of iterations for the `RS` starts
`k`	Number of clusters
`m`	Parameter of fuzziness
`ent`	Degree of fuzzy entropy (`NULL` for `NEFRC`)
`b`	Parameter of the polynomial fuzzifier (`NULL` for `NEFRC`)
`vp`	Volume parameter (`NULL` for `NEFRC`)
`delta`	Noise distance (`NULL` for `NEFRC`)
`stand`	Standardization (Yes if `stand=1`, No if `stand=0`) (`NULL` for `NEFRC`)
`Xca`	Data used in the clustering algorithm (`NULL` for `NEFRC`, `D` is used)
`X`	Raw data (`NULL` for `NEFRC`)
`D`	Dissimilarity matrix
`call`	Matched call

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Davé, R. N., & Sen, S. 2002. Robust fuzzy clustering of relational data. IEEE Transactions on Fuzzy Systems, 10(6), 713-727.

See Also

`NEFRC.noise`, `print.fclust`, `summary.fclust`, `plot.fclust`

Examples

```
## Not run:
require(cluster)
data("houseVotes")
X <- houseVotes[,-1]
D <- daisy(x = X, metric = "gower")
clust.NEFRC <- NEFRC(D = D, k = 2:6, m = 2, index = "SIL.F")
summary(clust.NEFRC)
plot(clust.NEFRC)

## End(Not run)
```

`NEFRC.noise` *Non-Euclidean Fuzzy Relational Clustering with noise cluster*

Description

Performs the Non-Euclidean Fuzzy Relational data Clustering algorithm.
The noise cluster is an additional cluster (with respect to the k standard clusters) such that objects recognized to be outliers are assigned to it with high membership degrees.

Usage

```
NEFRC.noise(D, k, m, delta, RS, startU, index, alpha, conv, maxit, seed)
```

Arguments

`D`	Matrix or data.frame containing distances/dissimilarities
`k`	An integer value or vector specifying the number of clusters for which the `index` is to be calculated (default: 2:6)
`m`	Parameter of fuzziness (default: 2)
`delta`	Noise distance (default: average observed distance)
`RS`	Number of (random) starts (default: 1)
`startU`	Rational start for the membership degree matrix `U` (default: no rational start)

index	Cluster validity index to select the number of clusters: "PC" (partition coefficient), "PE" (partition entropy), "MPC" (modified partition coefficient), "SIL" (silhouette), "SIL.F" (fuzzy silhouette) (default: "SIL.F")
alpha	Weighting coefficient for the fuzzy silhouette index SIL.F (default: 1)
conv	Convergence criterion (default: 1e-9)
maxit	Maximum number of iterations (default: 1e+6)
seed	Seed value for random number generation (default: NULL)

Details

If startU is given, the argument k is ignored (the number of clusters is ncol(startU)).
If startU is given, the first element of value, cput and iter refer to the rational start.

Value

Object of class fclust, which is a list with the following components:

U	Membership degree matrix
H	Prototype matrix (NULL for NEFRC.noise)
F	Array containing the covariance matrices of all the clusters (NULL for NEFRC.noise)
clus	Matrix containing the indexes of the clusters where the objects are assigned (column 1) and the associated membership degrees (column 2)
medoid	Vector containing the indexes of the medoid objects (NULL for NEFRC.noise)
value	Vector containing the loss function values for the RS starts
criterion	Vector containing the values of the cluster validity index
iter	Vector containing the numbers of iterations for the RS starts
k	Number of clusters
m	Parameter of fuzziness
ent	Degree of fuzzy entropy (NULL for NEFRC.noise)
b	Parameter of the polynomial fuzzifier (NULL for NEFRC.noise)
vp	Volume parameter (NULL for NEFRC.noise)
delta	Noise distance (NULL for NEFRC.noise).
stand	Standardization (Yes if stand=1, No if stand=0) (NULL for NEFRC.noise).
Xca	Data used in the clustering algorithm (NULL for NEFRC.noise), D is used)
X	Raw data (NULL for NEFRC.noise)
D	Dissimilarity matrix
call	Matched call

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Davé, R. N., & Sen, S. 2002. Robust fuzzy clustering of relational data. IEEE Transactions on Fuzzy Systems, 10(6), 713-727.

See Also

NEFRC, print.fclust, summary.fclust, plot.fclust

Examples

```
## Not run:
require(cluster)
data("houseVotes")
X <- houseVotes[,-1]
D <- daisy(x = X, metric = "gower")
clust.NEFRC.noise <- NEFRC.noise(D = D, k = 2:6, m = 2, index = "SIL.F")
summary(clust.NEFRC.noise)
plot(clust.NEFRC.noise)

## End(Not run)
```

PC *Partition coefficient*

Description

Produces the partition coefficient index. The optimal number of clusters k is is such that the index takes the maximum value.

Usage

```
PC (U)
```

Arguments

U	Membership degree matrix

Value

pc	Value of the partition coefficient index

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Bezdek J.C., 1974. Cluster validity with fuzzy sets. Journal of Cybernetics, 3, 58-73.

See Also

PE, MPC, SIL, SIL.F, XB, Fclust, Mc

Examples

```
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
## fuzzy k-means
## (excluded the factor column Type (last column))
clust=FKM(Mc[,1:(ncol(Mc)-1)],k=6,m=1.5,stand=1)
## partition coefficient
pc=PC(clust$U)
```

PE *Partition entropy*

Description

Produces the partition entropy index. The optimal number of clusters k is is such that the index takes the minimum value.

Usage

```
PE (U, b)
```

Arguments

U	Membership degree matrix
b	Logarithmic base (default: exp(1))

Value

pe	Value of the partition entropy index

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Bezdek J.C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York.

See Also

PC, MPC, SIL, SIL.F, XB, Fclust, Mc

Examples

```
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
## fuzzy k-means
## (excluded the factor column Type (last column))
clust=FKM(Mc[,1:(ncol(Mc)-1)],k=6,m=1.5,stand=1)
## partition entropy index
pe=PE(clust$U)
```

---

plot.fclust *Plotting fuzzy clustering output*

Description

Plot method for class fclust. The function creates a scatter plot visualizing the cluster structure. The objects are represented by points in the plot using observed variables or principal components.

Usage

```
## S3 method for class fclust
## S3 method for class 'fclust'
plot(x, v1v2, colclus, umin, ucex, pca, ...)
```

Arguments

x	Object of class fclust
v1v2	Vector with two elements specifying the numbers of the variables (or of the principal components) to be plotted (default: 1:2); in case of relational data, the argument is ignored
colclus	Vector specifying the color palette for the clusters (default: palette(rainbow(k)))
umin	Lowest maximal membership degree such that an object is assigned to a cluster (default: 0)
ucex	Logical value specifying if the points are magnified according to the maximal membership degree (if ucex=TRUE) (default: ucex=FALSE)
pca	Logical value specifying if the objects are represented using principal components (if pca=TRUE) (default: pca=FALSE); in case of relational data, the argument is ignored
...	Additional arguments arguments for plot

Details

In the scatter plot the objects are represented by circles (pch=16) and the prototypes by stars (pch=8) using observed variables (if pca=FALSE) or principal components (if pca=TRUE), the numbers of which are specified in v1v2. Their colors differ for every cluster according to colclus. Objects such that their maximal membership degrees are lower than umin are in black. The sizes of the circles depends on the maximal membership degrees of the corresponding objects if ucex=TRUE. Also note that principal components are extracted using standardized data.
In case of relational data, the first two components resulting from Non-metric Multidimensional Scaling performed using the package **MASS** are used.

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

See Also

VIFCR, VAT, VCV, VCV2, Fclust, print.fclust, summary.fclust, Mc

Examples

```
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
## fuzzy k-means
## (excluded the factor column Type (last column))
clust=FKM(Mc[,1:(ncol(Mc)-1)],k=6,m=1.5,stand=1)
## Scatter plot of Calories vs Cholesterol (mg)
names(Mc)
plot(clust,v1v2=c(1,5))
## Scatter plot of Calories vs Cholesterol (mg) using gray levels for the clusters
plot(clust,v1v2=c(1,5),colclus=gray.colors(6))
## Scatter plot of Calories vs Cholesterol (mg)
## coloring in black objects with maximal membership degree lower than 0.5
plot(clust,v1v2=c(1,5),umin=0.5)
## Scatter plot of Calories vs Cholesterol (mg)
## coloring in black objects with maximal membership degree lower than 0.5
## and magnifying the points according to the maximal membership degree
plot(clust,v1v2=c(1,5),umin=0.5,ucex=TRUE)
## Scatter plot using the first two principal components and
## coloring in black objects with maximal membership degree lower than 0.3
plot(clust,v1v2=1:2,umin=0.3,pca=TRUE)
```

print.fclust *Printing fuzzy clustering output*

Description

Print method for class `fclust`.

Usage

```
## S3 method for class fclust
## S3 method for class 'fclust'
print(x, ...)
```

Arguments

x	Object of class `fclust`
...	Additional arguments for `print`

Details

The function displays the number of objects, the number of clusters, the closest hard clustering partition (objects assigned to the clusters with the highest membership degree) and the membership degree matrix (rounded).

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

See Also

`Fclust`, `summary.fclust`, `plot.fclust`, `unemployment`

Examples

```
## unemployment data
data(unemployment)
## fuzzy k-means
unempFKM=FKM(unemployment,k=3,stand=1)
unempFKM
```

RI.F *Fuzzy Rand index*

Description

Produces the fuzzy version of the Rand index between a hard (reference) partition and a fuzzy partition.

Usage

```
RI.F(VC, U, t_norm)
```

Arguments

VC	Vector of class labels
U	Fuzzy membership degree matrix or data.frame
t_norm	Type of the triangular norm: "minimum" (minimum triangular norm), "triangular product" (product norm) (default: "minimum")

Value

`ri.f`Value of the fuzzy adjusted Rand index

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Campello, R.J., 2007. A fuzzy extension of the Rand index and other related indexes for clustering and classification assessment. Pattern Recognition Letters, 28, 833-841.
Rand, W.M., 1971. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association, 66, 846-850.

See Also

ARI.F, JACCARD.F, Fclust.compare

Examples

```
## Not run:
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
```

```
Mc=Mc[,-1]
## fuzzy k-means
## (excluded the factor column Type (last column))
clust=FKM(Mc[,1:(ncol(Mc)-1)],k=6,m=1.5,stand=1)
## fuzzy Rand index
ri.f=RI.F(VC=Mc$Type,U=clust$U)

## End(Not run)
```

SIL *Silhouette index*

Description

Produces the silhouette index. The optimal number of clusters k is is such that the index takes the maximum value.

Usage

```
SIL (Xca, U, distance)
```

Arguments

`Xca`	Matrix or data.frame
`U`	Membership degree matrix
`distance`	If `distance=TRUE`, `Xca` is assumed to contain distances/dissimilarities (default: FALSE)

Details

`Xca` should contain the same dataset used in the clustering algorithm, i.e., if the clustering algorithm is run using standardized data, then `SIL` should be computed using the same standardized data.
Set `distance=TRUE` if `Xca` is a distance/dissimilarity matrix.

Value

`sil.obj`	Vector containing the silhouette indexes for all the objects
`sil`	Value of the silhouette index (mean of `sil.obj`)

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Kaufman L., Rousseeuw P.J., 1990. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York.

See Also

PC, PE, MPC, SIL.F, XB, Fclust, Mc

Examples

```
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
## fuzzy k-means
## (excluded the factor column Type (last column))
clust=FKM(Mc[,1:(ncol(Mc)-1)],k=6,m=1.5,stand=1)
## silhouette index
sil=SIL(clust$Xca,clust$U)
```

SIL.F *Fuzzy silhouette index*

Description

Produces the fuzzy silhouette index. The optimal number of clusters *k* is is such that the index takes the maximum value.

Usage

```
SIL.F (Xca, U, alpha, distance)
```

Arguments

Xca	Matrix or data.frame
U	Membership degree matrix
alpha	Weighting coefficient (default: 1)
distance	If `distance=TRUE`, `Xca` is assumed to contain distances/dissimilarities (default: FALSE)

Details

`Xca` should contain the same dataset used in the clustering algorithm, i.e., if the clustering algorithm is run using standardized data, then `SIL.F` should be computed using the same standardized data. Set `distance=TRUE` if `Xca` is a distance/dissimilarity matrix.

Value

sil.f	Value of the fuzzy silhouette index

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Campello R.J.G.B., Hruschka E.R., 2006. A fuzzy extension of the silhouette width criterion for cluster analysis. Fuzzy Sets and Systems, 157, 2858-2875.

See Also

PC, PE, MPC, SIL, XB, Fclust, Mc

Examples

```
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
## fuzzy k-means
## (excluded the factor column Type (last column))
clust=FKM(Mc[,1:(ncol(Mc)-1)],k=6,m=1.5,stand=1)
## fuzzy silhouette index
sil.f=SIL.F(clust$Xca,clust$U)
```

---

summary.fclust — *Summarizing fuzzy clustering output*

Description

Summary method for class `fclust`.

Usage

```
## S3 method for class fclust
## S3 method for class 'fclust'
summary(object, ...)
```

Arguments

`object`	Object of class `fclust`
`...`	Additional arguments for `summary`

Details

The function displays the number of objects, the number of clusters, the cluster sizes, the closest hard clustering partition (objects assigned to the clusters with the highest membership degree), the cluster memberships (using the closest hard clustering partition), the number of objects with unclear assignment (when the maximal membership degree is lower than 0.5), the objects with unclear assignment and the cluster sizes without unclear assignments (only if objects with unclear assignment are present), the cluster summary (for every cluster: size, minimal membership degree, maximal membership degree, average membership degree, number of objects with unclear assignment) and the Euclidean distance matrix for the cluster prototypes.

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

See Also

`Fclust`, `print.fclust`, `plot.fclust`, `unemployment`

Examples

```
## unemployment data
data(unemployment)
## fuzzy k-means
unempFKM=FKM(unemployment,k=3,stand=1)
summary(unempFKM)
```

`synt.data` *Synthetic data*

Description

Synthetic dataset with 2 non-spherical clusters.

Usage

```
data(synt.data)
```

Format

A matrix with 302 rows and 2 columns.

Details

Although two clusters are clearly visible, fuzzy k-means fails to discover them. The Gustafson and Kessel-like fuzzy k-means should be used for finding the known-in-advance clusters.

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

See Also

Fclust, FKM, FKM.gk, plot.fclust

Examples

```
## Not run:
## synthetic data
data(synt.data)
plot(synt.data)
## fuzzy k-means
syntFKM=FKM(synt.data)
## Gustafson and Kessel-like fuzzy k-means
syntFKM.gk=FKM.gk(synt.data)
## plot of cluster structures from fuzzy k-means and Gustafson and Kessel-like fuzzy k-means
par(mfcol = c(2,1))
plot(syntFKM)
plot(syntFKM.gk)

## End(Not run)
```

---

synt.data2 *Synthetic data*

Description

Synthetic dataset with 2 non-spherical clusters.

Usage

```
data(synt.data2)
```

Format

A matrix with 240 rows and 2 columns.

Details

Although three clusters are clearly visible, Gustafson and Kessel - like fuzzy *k*-means clustering algorithm `FKM.gk` fails due to singularity of some covariance matrix. The Gustafson, Kessel and Babuska - like fuzzy *k*-means clustering algorithm `FKM.gkb` should be used to avoid singularity problem.

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Gustafson E.E., Kessel W.C., 1978. Fuzzy clustering with a fuzzy covariance matrix. Proceedings of the IEEE Conference on Decision and Control, pp. 761-766.

See Also

Fclust, FKM.gk, FKM.gkb, plot.fclust

Examples

```
data(synt.data2)
plot(synt.data2)

## Gustafson and Kessel-like fuzzy k-means
syntFKM.gk=FKM.gk(synt.data2, k = 3, RS = 1, seed = 123)
## Gustafson, Kessel and Babuska-like fuzzy k-means
syntFKM.gkb=FKM.gkb(synt.data2, k = 3, RS = 1, seed = 123)
```

---

unemployment *Unemployment data*

Description

Unemployment data about some European countries in 2011.

Usage

```
data(unemployment)
```

Format

A data.frame with 32 rows and 3 columns.

Details

The source is Eurostat news-release 104/2012 - 4 July 2012. The 32 observations are European countries: BELGIUM, BULGARIA, CZECHREPUBLIC, DENMARK, GERMANY, ESTONIA, IRELAND, GREECE, SPAIN, FRANCE, ITALY, CYPRUS, LATVIA, LITHUANIA, LUXEMBOURG, HUNGARY, MALTA, NETHERLANDS, AUSTRIA, POLAND, PORTUGAL, ROMANIA, SLOVENIA, SLOVAKIA, FINLAND, SWEDEN, UNITEDKINGDOM, ICELAND, NORWAY, SWITZERLAND, CROATIA, TURKEY. The 3 variables are: the total unemployment rate, defined as the percentage of unemployed persons aged 15-74 in the economically active population (Variable 1); the youth unemployment rate, defined as the unemployment rate for young people aged between 15 and 24 (Variable 2); the long-term unemployment share, defined as the Percentage of unemployed persons who have been unemployed for 12 months or more (Variable 3). Non-spherical clusters seem to be present in the data. The Gustafson and Kessel-like fuzzy k-means should be used for finding them.

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

See Also

Fclust, FKM, FKM.gk

Examples

```
## unemployment data
data(unemployment)
## fuzzy k-means (only spherical clusters)
unempFKM=FKM(unemployment,k=3)
## Gustafson and Kessel-like fuzzy k-means (non-spherical clusters)
unempFKM.gk=FKM.gk(unemployment,k=3,RS=10)
```

VAT — *Visual Assessment of (Cluster) Tendency*

Description

Digital intensity image to inspect the number of clusters

Usage

```
VAT (Xca)
```

Arguments

Xca	Matrix or data.frame (usually data to be used in the clustering algorithm)

Details

Each cell refers to a dissimilarity between a pair of objects. Small dissimilarities are represented by dark shades and large dissimilarities are represented by light shades. In the plot the dissimilarities are reorganized in such a way that, roughly speaking, (darkly shaded) diagonal blocks correspond to clusters in the data. Therefore, *k* dark blocks along its main diagonal suggest that the data contain *k* (as yet unfound) clusters and the size of each block represents the approximate size of the cluster.

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Bezdek J.C., Hathaway, R.J., 2002. VAT: a tool for visual assessment of (cluster) tendency. Proceedings of the IEEE International Joint Conference on Neural Networks, , pp. 2225?2230.
Hathaway R.J., Bezdek J.C., 2003. Visual cluster validity for prototype generator clustering models. Pattern Recognition Letters, 24, 1563?1569.
Huband J.M., Bezdek J.C., 2008. VCV2 ? Visual Cluster Validity. In Zurada J.M., Yen G.G., Wang J. (Eds.): Lecture Notes in Computer Science, 5050, pp. 293?308. Springer-Verlag, Berlin Heidelberg.

See Also

plot.fclust, VIFCR, VCV, VCV2, Mc

Examples

```
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## data standardization (after removing the column Serving Size)
Mc=scale(Mc[,1:(ncol(Mc)-1)],center=TRUE,scale=TRUE)[,]
## plot of VAT
VAT(Mc)
```

VCV *Visual Cluster Validity*

Description

Digital intensity image generated using the prototype matrix (and the membership degree matrix) to do cluster validation. The function also plots the VAT image.

Usage

```
VCV (Xca, U, H, which)
```

Arguments

Xca	Matrix or data.frame (usually data used in the clustering algorithm)
U	Membership degree matrix
H	Prototype matrix
which	If a subset of the plots is required, specify a subset of the numbers `1:2` (default: `1:2`)

.

Details

Plot 1 (`which=1`): VAT. Each cell refers to a dissimilarity between a pair of objects. Small dissimilarities are represented by dark shades and large dissimilarities are represented by light shades. In the plot the dissimilarities are reorganized in such a way that, roughly speaking, (darkly shaded) diagonal blocks correspond to clusters in the data. Therefore, k dark blocks along its main diagonal suggest that the data contain k (as yet unfound) clusters and the size of each block represents the approximate size of the cluster.
Plot 2 (`which=2`): VCV. Each cell refers to a dissimilarity between a pair of objects computed with respect to the cluster prototypes. Small dissimilarities are represented by dark shades and large

dissimilarities are represented by light shades. In the plot the dissimilarities are organized by reordering the clusters (the original first cluster is the first reordered cluster and the remaining clusters are reordered so that (new) cluster c+1 is the nearest of the remaining clusters to (newly indexed) cluster c) and the objects (in accordance with decreasing membership degrees). If k dark blocks along its main diagonal are visible, then a k-cluster structure is revealed. Note that the actual number of clusters can be revealed even when a larger number of clusters is used. This suggests that the correct value of k can sometimes be found by running the algorithm with a large value of k, and then ascertaining its correct value from the visual evidence in the VCV image.

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Bezdek J.C., Hathaway, R.J., 2002. VAT: a tool for visual assessment of (cluster) tendency. Proceedings of the IEEE International Joint Conference on Neural Networks, , pp. 2225?2230.

Hathaway R.J., Bezdek J.C., 2003. Visual cluster validity for prototype generator clustering models. Pattern Recognition Letters, 24, 1563?1569.

See Also

plot.fclust, VIFCR, VAT, VCV2, Mc

Examples

```
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
## fuzzy k-means
## (excluded the factor column Type (last column))
clust=FKM(Mc[,1:(ncol(Mc)-1)],k=6,m=1.5,stand=1)
## plots of VAT and VCV
VCV(clust$Xca,clust$U,clust$H)
## plot of VCV
VCV(clust$Xca,clust$U,clust$H, 2)
```

VCV2 *(New) Visual Cluster Validity*

Description

Digital intensity image generated using the membership degree matrix to do cluster validation. The function also plots the VAT image.

Usage

```
VCV2 (Xca, U, which)
```

Arguments

`Xca`	Matrix or data.frame (usually data used in the clustering algorithm)
`U`	Membership degree matrix
`which`	If a subset of the plots is required, specify a subset of the numbers `1:2` (default: `1:2`)

.

Details

Plot 1 (`which=1`): VAT. Each cell refers to a dissimilarity between a pair of objects. Small dissimilarities are represented by dark shades and large dissimilarities are represented by light shades. In the plot the dissimilarities are reorganized in such a way that, roughly speaking, (darkly shaded) diagonal blocks correspond to clusters in the data. Therefore, k dark blocks along its main diagonal suggest that the data contain k (as yet unfound) clusters and the size of each block represents the approximate size of the cluster.
Plot 2 (`which=2`): VCV2. Each cell refers to a dissimilarity between a pair of objects computed with respect to the cluster membership degrees. Small dissimilarities are represented by dark shades and large dissimilarities are represented by light shades. In the plot the dissimilarities are reorganized by using the VAT reordering. If k dark blocks along its main diagonal are visible, then a k-cluster structure is revealed. Note that the actual number of clusters can be revealed even when a larger number of clusters is used. This suggests that the correct value of k can sometimes be found by running the algorithm with a large value of k, and then ascertaining its correct value from the visual evidence in the VCV2 image.

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Bezdek J.C., Hathaway, R.J., 2002. VAT: a tool for visual assessment of (cluster) tendency. Proceedings of the IEEE International Joint Conference on Neural Networks, , pp. 2225?2230.
Huband J.M., Bezdek J.C., 2008. VCV2 ? Visual Cluster Validity. In Zurada J.M., Yen G.G., Wang J. (Eds.): Lecture Notes in Computer Science, 5050, pp. 293?308. Springer-Verlag, Berlin Heidelberg.

See Also

`plot.fclust`, `VIFCR`, `VAT`, `VCV`, `Mc`

Examples

```
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
## fuzzy k-means
## (excluded the factor column Type (last column))
clust=FKM(Mc[,1:(ncol(Mc)-1)],k=6,m=1.5,stand=1)
## plots of VAT and VCV2
VCV2(clust$Xca,clust$U)
## plot of VCV2
VCV2(clust$Xca,clust$U, 2)
```

VIFCR *Visual inspection of fuzzy clustering results*

Description

Plots for validation of fuzzy clustering results. Three plots (selected by `which`) are available.

Usage

```
VIFCR (fclust.obj, which)
```

Arguments

`fclust.obj`	Object of class `fclust`
`which`	If a subset of the plots is required, specify a subset of the numbers `1:3` (default: `1:3`)

.

Details

Plot 1 (`which=1`). Histogram of the membership degrees setting `breaks=seq(from=0,to=1,by=0.1)`. The frequencies are scaled so that the heights of the first and the latter rectangles are the same in the ideal case of crisp (non-fuzzy) memberships. The fuzzy clustering solution should be such that the heights of the first and the latter rectangles are high and those of the rectangles in the middle are low. High heights of rectangles in the middle denote the presence of ambiguous membership degrees. This is an indicator for a non-optimal clustering result.
Plot 2 (`which=2`). Scatter plot of the objects at the co-ordinates (u1,u2). For each object, u1 and u2 denote, respectively, the highest and the second highest membership degrees. All points lie within the triangle with vertices (0,0), (0.5,0.5) and (1,0). In the ideal case of (almost) crisp membership degrees all points are near the vertex (1,0). Points near the vertex (0.5,0.5) highlight ambiguous

objects shared by two clusters. Points near the vertex (0,0) are usually outliers characterized by low membership degrees to all clusters (provided that the noise approach is considered).
Plot 3 (`which=3`). For each cluster, scatter plot of the of the objects at the co-ordinates (dc,uc). For each object, dc is the squared Euclidean distance between the object and the cluster prototype and uc is the membership degree of the object to the cluster. The ideal case is such that points are in the upper left area or in the lower right area. In fact, this highlights high membership degrees for small distances and low membership degrees for large distances.

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Klawonn F., Chekhtman V., Janz E., 2003. Visual inspection of fuzzy clustering results. In Benitez J.M., Cordon O., Hoffmann, F., Roy R. (Eds.): Advances in Soft Computing - Engineering Design and Manufacturing, pp. 65-76. Springer, London.

See Also

`plot.fclust`, `VAT`, `VCV`, `VCV2`, `unemployment`

Examples

```
## unemployment data
data(unemployment)
## fuzzy k-means
unempFKM=FKM(unemployment,k=3,stand=1)
## all plots
VIFCR(unempFKM)
## plots 1 and 3
VIFCR(unempFKM,c(1,3))
```

`XB` *Xie and Beni index*

Description

Produces the Xie and Beni index. The optimal number of clusters k is is such that the index takes the minimum value.

Usage

```
XB (Xca, U, H, m)
```

Arguments

Xca	Matrix or data.frame
U	Membership degree matrix
H	Prototype matrix
m	Parameter of fuzziness (default: 2)

Details

Xca should contain the same dataset used in the clustering algorithm, i.e., if the clustering algorithm is run using standardized data, then XB should be computed using the same standardized data.
m should be the same parameter of fuzziness used in the clustering algorithm.

Value

xb	Value of the Xie and Beni index

Author(s)

Paolo Giordani, Maria Brigida Ferraro, Alessio Serafini

References

Xie X.L., Beni G. (1991). A validity measure for fuzzy clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13, 841-847.

See Also

PC, PE, MPC, SIL, SIL.F, Fclust, Mc

Examples

```
## McDonald's data
data(Mc)
names(Mc)
## data normalization by dividing the nutrition facts by the Serving Size (column 1)
for (j in 2:(ncol(Mc)-1))
Mc[,j]=Mc[,j]/Mc[,1]
## removing the column Serving Size
Mc=Mc[,-1]
## fuzzy k-means
## (excluded the factor column Type (last column))
clust=FKM(Mc[,1:(ncol(Mc)-1)],k=6,m=1.5,stand=1)
## Xie and Beni index
xb=XB(clust$Xca,clust$U,clust$H,clust$m)
```

Index

∗ **datasets**
synt.data2, 71
∗ **data**
butterfly, 4
houseVotes, 50
Mc, 54
NBA, 56
synt.data, 70
unemployment, 72
∗ **multivariate**
ARI.F, 3
butterfly, 4
cl.memb, 5
cl.memb.H, 6
cl.memb.t, 7
cl.size, 8
cl.size.H, 9
Fclust, 10
Fclust.compare, 11
Fclust.index, 13
FKM, 14
FKM.ent, 16
FKM.ent.noise, 18
FKM.gk, 20
FKM.gk.ent, 23
FKM.gk.ent.noise, 25
FKM.gk.noise, 28
FKM.gkb, 30
FKM.gkb.ent, 32
FKM.gkb.ent.noise, 34
FKM.gkb.noise, 37
FKM.med, 39
FKM.med.noise, 42
FKM.noise, 44
FKM.pf, 46
FKM.pf.noise, 48
houseVotes, 50
Hraw, 51
JACCARD.F, 52
Mc, 54
MPC, 55
NBA, 56
NEFRC, 57
NEFRC.noise, 59
PC, 61
PE, 62
plot.fclust, 63
print.fclust, 65
RI.F, 66
SIL, 67
SIL.F, 68
summary.fclust, 69
synt.data, 70
unemployment, 72
VAT, 73
VCV, 74
VCV2, 75
VIFCR, 77
XB, 78

ARI.F, 3, *12*, *53*, *66*

butterfly, 4, *20*, *43*, *45*

cl.memb, 5, *6*, *7*
cl.memb.H, *5*, 6, *7*
cl.memb.t, *5*, *6*, 7
cl.size, 8, *9*
cl.size.H, *8*, 9

Fclust, *4*, 10, *13*, *15*, *18*, *20*, *22*, *25*, *27*, *29*, *32*, *34*, *36*, *39*, *41*, *43*, *45*, *47*, *50*, *52*, *54*, *56*, *62–65*, *68–73*, *79*
Fclust.compare, *3*, *10*, 11, *53*, *66*
Fclust.index, *10*, 13, *15*, *18*, *20*, *22*, *25*, *27*, *29*, *32*, *34*, *36*, *39*, *41*, *43*, *45*, *47*, *50*
FKM, *4*, *10*, 14, *45*, *54*, *57*, *71*, *73*
FKM.ent, *10*, 16, *20*, *54*
FKM.ent.noise, *10*, *18*, 18

FKM.gk, *10*, 20, *32*, *71–73*
FKM.gk.ent, *10*, 23, *34*
FKM.gk.ent.noise, *10*, 25, *36*
FKM.gk.noise, *10*, 28, *39*
FKM.gkb, *10*, *22*, 30, *72*
FKM.gkb.ent, *10*, *25*, 32
FKM.gkb.ent.noise, *10*, *27*, 34
FKM.gkb.noise, *10*, *29*, 37
FKM.med, *10*, 39, *43*, *54*
FKM.med.noise, *10*, *41*, 42
FKM.noise, *4*, *10*, *15*, 44
FKM.pf, *10*, 46, *50*
FKM.pf.noise, *10*, *47*, 48

houseVotes, 50
Hraw, 51

JACCARD.F, *3*, *12*, 52, *66*

Mc, *13*, *15*, *18*, *41*, *47*, *50*, 54, *56*, *62–64*, *68*, *69*, *74–76*, *79*
MPC, *13*, 55, *62*, *63*, *68*, *69*, *79*

NBA, 56
NEFRC, *10*, *51*, 57, *61*
NEFRC.noise, *10*, *51*, *59*, 59

PC, *13*, *56*, 61, *63*, *68*, *69*, *79*
PE, *13*, *56*, *62*, 62, *68*, *69*, *79*
plot, *63*
plot.fclust, *10*, *15*, *18*, *20*, *22*, *25*, *27*, *29*, *32*, *34*, *36*, *39*, *41*, *43*, *45*, *47*, *50*, *59*, *61*, 63, *65*, *70–72*, *74–76*, *78*
print, *65*
print.fclust, *10*, *15*, *18*, *20*, *22*, *25*, *27*, *29*, *32*, *34*, *36*, *39*, *41*, *43*, *45*, *47*, *50*, *59*, *61*, *64*, 65, *70*

RI.F, *3*, *12*, *53*, 66

SIL, *13*, *56*, *62*, *63*, 67, *69*, *79*
SIL.F, *13*, *56*, *62*, *63*, *68*, 68, *79*
summary, *69*
summary.fclust, *10*, *15*, *18*, *20*, *22*, *25*, *27*, *29*, *32*, *34*, *36*, *39*, *41*, *43*, *45*, *47*, *50*, *59*, *61*, *64*, *65*, 69
synt.data, 70
synt.data2, 71

unemployment, *22*, *25*, *27*, *29*, *32*, *34*, *36*, *39*, *52*, *65*, *70*, 72, *78*

VAT, *64*, 73, *75*, *76*, *78*
VCV, *64*, *74*, 74, *76*, *78*
VCV2, *64*, *74*, *75*, 75, *78*
VIFCR, *64*, *74–76*, 77

XB, *13*, *56*, *62*, *63*, *68*, *69*, 78