

Package ‘pfclust’

April 28, 2026

Type Package

Title Power Fuzzy Clustering and Cluster-Wise Regression

Version 0.1.0

Description Implementations of Power Fuzzy Clustering (PFC) and Power Fuzzy Cluster-wise Regression (PFCR) for multivariate data. The package supports Minkowski distances, with the L1 case solved via iteratively re-weighted least squares and the case $p > 1$ solved via coordinate-wise root finding, as well as an adaptive, regularised Mahalanobis distance with per-cluster covariance matrices. Both plain fuzzy clustering and cluster-wise linear regression are provided. The corresponding paper can be found at Nguyen P.T., Tortora C., and Punzo A. (2026) <[doi:10.1109/TFUZZ.2026.3683998](https://doi.org/10.1109/TFUZZ.2026.3683998)>.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.5.0)

Imports stats

Suggests flexCWM

RoxygenNote 7.3.3

NeedsCompilation no

Author Phuc Thinh Nguyen [aut, cre],
Cristina Tortora [aut, ths, dgs],
Antonio Punzo [aut, ths, dgs]

Maintainer Phuc Thinh Nguyen <phucthinh010603@yahoo.com>

Repository CRAN

Date/Publication 2026-04-28 18:20:02 UTC

Contents

PFC	2
PFCR	3
Index	6

Description

Clusters the rows of Y into K groups using Minkowski or adaptive regularised Mahalanobis distance.

Usage

```
PFC(
  Y,
  K,
  m = 2,
  q = 2,
  distance = "Minkowski",
  p = 2,
  alpha = 0.5,
  beta = 10^15,
  threshold = 0.01,
  max.iter = 100
)
```

Arguments

Y	An $n \times d$ data frame or matrix of observations.
K	Number of clusters (positive integer).
m	Fuzzifier, must be strictly greater than 1. Default 2.
q	Distance exponent, must be strictly greater than 0. Default 2.
<code>distance</code>	Either "Minkowski" or "Mahalanobis".
p	Minkowski exponent (≥ 1). Ignored when <code>distance = "Mahalanobis"</code> . Default 2.
<code>alpha</code>	Regularisation weight for the Mahalanobis covariance. Default 0.5.
<code>beta</code>	Eigenvalue ratio bound for the Mahalanobis covariance. Default $1e15$.
<code>threshold</code>	Convergence tolerance. Default 0.01.
<code>max.iter</code>	Maximum number of iterations. Default 100.

Value

A list with elements B (or C) for cluster centres, d (distances), p (memberships), JDF (objective history), and l (hard labels). For Mahalanobis, also ρ and cov .

References

- P.T. Nguyen, C. Tortora, and A. Punzo. (2026) "Power fuzzy clustering: flexible distance metrics and inclusion of covariates". IEEE Transactions on Fuzzy Systems 10.1109/TFUZZ.2026.3683998
- D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in 1978 IEEE conference on decision and control including the 17th symposium on adaptive processes. IEEE, 1979, pp. 761–766.

Examples

```
Y <- iris[, 1:4]
l <- iris[, 5]

# --- 2a. Mahalanobis distance ---
resMahC <- PFC(Y, K = 3, distance = "Mahalanobis")
table(resMahC$l, l)
pairs(Y, col = resMahC$l)

# --- 2b. Minkowski distance (default p=2) ---
resMinC <- PFC(Y, K = 3)
table(resMinC$l, l)
pairs(Y, col = resMinC$l)

# --- 2c. Minkowski distance (p=1, Manhattan) ---
resMin1C <- PFC(Y, K = 3, p = 1)
table(resMin1C$l, l)
pairs(Y, col = resMin1C$l)
```

PFCR

Power Fuzzy Cluster-wise Regression

Description

Fits K cluster-specific linear models $Y = XB_k + \varepsilon$, selecting an internal solver based on the chosen distance and exponent.

Usage

```
PFCR(
  X,
  Y,
  K,
  m = 2,
  q = 2,
  distance = "Minkowski",
  p = 2,
  alpha = 0.5,
  beta = 10^15,
```

```

    threshold = 0.01,
    max.iter = 100
  )

```

Arguments

<code>X</code>	An $n \times dx$ data frame or matrix of covariates.
<code>Y</code>	An $n \times dy$ data frame or matrix of dependent variables.
<code>K</code>	Number of clusters (positive integer).
<code>m</code>	Fuzzifier, must be strictly greater than 1. Default 2.
<code>q</code>	Distance exponent, must be strictly greater than 0. Default 2.
<code>distance</code>	Either "Minkowski" or "Mahalanobis".
<code>p</code>	Minkowski exponent (≥ 1). Ignored when <code>distance = "Mahalanobis"</code> . Default 2.
<code>alpha</code>	Regularisation weight for the Mahalanobis covariance. Default 0.5.
<code>beta</code>	Eigenvalue ratio bound for the Mahalanobis covariance. Default $1e15$.
<code>threshold</code>	Convergence tolerance on successive coefficient updates. Default 0.01.
<code>max.iter</code>	Maximum number of iterations. Default 100.

Value

A list with elements:

- B** Array of regression coefficients.
- d** Data frame of distances ($n \times K$).
- p** Data frame of membership degrees ($n \times K$).
- JDF** Vector of objective-function values per iteration.
- l** Hard cluster labels (length n).
- rho, cov** (Mahalanobis only) cluster proportions and covariance matrices.

References

P.T. Nguyen, C. Tortora, and A. Punzo. (2026) "Power fuzzy clustering: flexible distance metrics and inclusion of covariates". *IEEE Transactions on Fuzzy Systems* 10.1109/TFUZZ.2026.3683998

Examples

```

library(flexCWM)
data("students")
Y <- students[, 2:3]
X <- students[, 4]
l <- students[, 1]

# --- 1a. Mahalanobis distance (default m=2, q=2) ---
resMah <- PFCR(X, Y, K = 2, distance = "Mahalanobis")

```

```
table(resMah$l, 1)

# --- 1b. Minkowski distance (default m=2, q=2, p=2) ---
resMin <- PFCR(X, Y, K = 2)
table(resMin$l, 1)

# --- 1c. Mahalanobis with m=3, q=3 ---
resMah33 <- PFCR(X, Y, K = 2, m = 3, q = 3, distance = "Mahalanobis")
table(resMah33$l, 1)

# --- Plots for the (m=3, q=3) Mahalanobis result ---
pp <- apply(resMah33$p, 1, max)
color <- ifelse(resMah33$l == unique(resMah33$l)[1], "blue", "red")

plot(Y, col = color, pch = 16, cex = pp,
      main = "PFCR Mahalanobis (3,3)")

plot(X, Y[, 1], ylab = "HEIGHT", xlab = "HEIGHT.F",
      pch = 16, cex = pp, col = color,
      main = "PFCR Mahalanobis (3,3)")
abline(resMah33$B[1, 1, 1], resMah33$B[2, 1, 1], col = "blue", lwd = 2)
abline(resMah33$B[1, 1, 2], resMah33$B[2, 1, 2], col = "red", lwd = 2)

plot(X, Y[, 2], ylab = "WEIGHT", xlab = "HEIGHT.F",
      pch = 16, cex = pp, col = color,
      main = "PFCR Mahalanobis (3,3)")
abline(resMah33$B[1, 2, 1], resMah33$B[2, 2, 1], col = "blue", lwd = 2)
abline(resMah33$B[1, 2, 2], resMah33$B[2, 2, 2], col = "red", lwd = 2)
```

Index

PFC, [2](#)
PFCR, [3](#)