Package: bootkmeans (via r-universe)

May 14, 2026

Type Package

Title A Bootstrap Augmented k-Means Algorithm for Fuzzy Partitions

Version 1.0.0

Date 2025-09-18

Author Jesse S. Ghashti [aut, cre], Jeffrey L. Andrews [aut], John R.J. Thompson [aut], Joyce Epp [aut], Harkunwar S. Kochar [aut]

Maintainer Jesse S. Ghashti <jesse.ghashti@ubc.ca>

Description Implementation of the bootkmeans algorithm, a bootstrap augmented k-means algorithm that returns probabilistic cluster assignments. From paper by Ghashti, J.S., Andrews, J.L. Thompson, J.R.J., Epp, J. and H.S. Kochar (2025), ``A bootstrap augmented k-means algorithm for fuzzy partitions" (Submitted).

License GPL-2

Encoding UTF-8

Depends R (>= 3.5.0), lmtest, abind

Imports MASS, stats, fclust, Thresher, mvtnorm

Suggests knitr, markdown, ggplot2, patchwork, scales, spelling

VignetteBuilder knitr

NeedsCompilation no

Language en-US

Repository https://ghashti-j.r-universe.dev

Date/Publication 2025-10-10 21:47:39 UTC

RemoteUrl https://github.com/ghashti-j/bootkmeans

RemoteRef HEAD

RemoteSha 52062b92a2f5187390ba42bb7ed0857c3feb325f

Contents

`boot.kmeans` *Bootstrap augmented k-means algorithm for fuzzy partitions*

Description

Repeatedly bootstraps the rows of a data matrix, runs `kmeans` on each resample (with optional seeding for given centres), tracks per-observation allocations using squared Euclidean distance, and aggregates results into out-of-bag (OOB) fuzzy memberships, hard clusters, and averaged cluster centres. Iterations can stop adaptively using a serial-correlation test on the objective trace.

Usage

```
boot.kmeans(
  data = NULL,
  groups = NULL,
  iterations = 500,
  nstart = 1,
  export = FALSE,
  display = FALSE,
  pval = 0.05,
  itermax = 10,
  maxsamp = 1000,
  verbose = FALSE,
  returnall = FALSE
)
```

Arguments

`data`	Numeric matrix or data frame of row observations and column variables. Required.
`groups`	Either and integer number of clusters K; or a $K \times p$ numeric matrix of initial centres. Required.
`iterations`	Initial number of bootstrap iterations to run before considering stopping (`default = 500`).
`nstart`	Passed to `kmeans` when `groups` is an integer (number of random starts, `default = 1`).

export	Logical; if TRUE, saves a JPEG of the objective trace at each iteration (plot<i>.jpg). Defaults to FALSE.
display	Logical; if TRUE, plots the most recent objective values during fitting. Defaults to FALSE.
pval	Significance threshold for adaptive stopping. When the Breusch–Godfrey test p-value on the last iterations objective values is not below pval, the procedure stops.
itermax	Maximum number of iterations per k-means run (passed to kmeans(iter.max = ...)).
maxsamp	Upper bound on total iterations if adaptive stopping keeps extending (default = 1000).
verbose	Logical; if TRUE, print iteration counter and latest test p-value while running. Defaults to FALSE.
returnall	Logical; if TRUE, return full per-iteration objects (centres, k-means fits, OOB lists); otherwise a smaller object of final results if return. Defaults to TRUE.

Details

Each iteration draws a bootstrap sample of rows, runs kmeans on the resample (first using either supplied centres or nstart random starts; subsequent iterations use the previous iteration's centres), and computes squared Euclidean distances from every original observation to each current centre using mahalanobis with the identity covariance. Observations are allocated to their nearest centre and these allocations are tracked across iterations.

Out-of-bag (OOB) sets are the observations note included in a given bootstrap sample. For each observation, its OOB allocations across the most recent iterations runs are tallied to produce a fuzzy membership matrix (U) and a hard label by maximum membership.

Convergence is assessed adaptively: on the trace of summed per-observation minimum squared distances (the k-means objective) over the most recent iterations runs, a Breusch–Godfrey serial-correlation test (bgtest applied to a regression of the objective on iteration index) is computed. If the p-value is below pval and iterations < maxsamp, one more iteration is added; otherwise the loop terminates. Final centres are the elementwise mean of the centres over the last iterations runs.

Value

An object of class "BSKMeans": a list with components

U	$n \times K$ matrix of OOB fuzzy cluster memberships.
clusters	Integer vector of length n of hard cluster labels.
centres	$K \times p$ matrix of averaged centres over the last iterations runs.
p.value	Final Breusch–Godfrey test p-value used for stopping.
iterations	Total number of iterations actually run.
occurences	$n \times$ iterations matrix of per-iteration allocations for all observations.
size	Number of clusters K.
soslist	Numeric vector of objective values by iteration.

centrelist	(If returnall = TRUE) list of per-iteration centre matrices; otherwise NULL.
ooblist	(If returnall = TRUE) list of OOB index vectors by iteration; otherwise NULL.
kmlist	(If returnall = TRUE) list of kmeans fit objects by iteration; otherwise NULL.

Author(s)

Jesse S. Ghashti <jesse.ghashti@ubc.ca> and Jeffrey L. Andrews <jeff.andrews@ubc.ca>

References

Ghashti, J.S., Andrews, J.L., Thompson, J.R.J., Epp, J. and H.S. Kochar (2025). A bootstrap augmented k-means algorithm for fuzzy partitions. Submitted.

Breusch, T.S. (1978). Testing for Autocorrelation in Dynamic Linear Models, *Australian Economic Papers*, 17, 334-355.

Godfrey, L.G. (1978). Testing Against General Autoregressive and Moving Average Error Models when the Regressors Include Lagged Dependent Variables', *Econometrica*, 46, 1293-1301.

See Also

compare.clusters, compare.tables, bootk.hardsoftvis, kmeans, bgtest

Examples

```
set.seed(1)

# basic usage
x <- as.matrix(iris[, -5])
fit <- boot.kmeans(data = x, groups = 3, iterations = 50, itermax = 20, verbose = TRUE)
table(fit$clusters, iris$Species)

# basic usage with initial cluster centres supplied
centres.init <- x[sample(nrow(x), 3), ]
fit2 <- boot.kmeans(data = x, groups = centres.init, iterations = 50)

# plot objective trace
plot(fit$soslist, type = "l", xlab = "Iteration", ylab = "Objective Function Value")
```

---

bootk.hardsoftvis *Visualize hard vs. soft assignments from bootstrap k-means*

Description

Plots the results of boot.kmeans highlighting which observations are assigned with full certainty (hard) versus fractional out-of-bag membership (soft/fuzzy). Either produces a full scatterplot matrix using all variables or a 2D scatterplot of chosen variables.

Usage

```
bootk.hardsoftvis(data = NULL, res, plotallvars = FALSE, var1 = NULL, var2 = NULL)
```

Arguments

data	Numeric data frame or matrix used for clustering in boot.kmeans. Required.
res	Result list returned from boot.kmeans (an object of class "BSKMeans").
plotallvars	Logical; if TRUE, plot all pairwise scatterplots via pairs, otherwise FALSE requires var1 and var2 arguments for a 2D scatterplot. Default FALSE.
var1	Integer column number for the x-axis variable when plotallvars = FALSE.
var2	Integer column number for the y-axis variable when plotallvars = FALSE.

Details

Each observation is classified as *hard* if any entry of its membership row U[i,] is exactly 1, and *soft* otherwise. These categories are mapped to colors green for hard assignments, blue for soft/fuzzy. With plotallvars = TRUE, a scatterplot matrix of all variables is drawn. With plotallvars = FALSE, only the two specified variables are plotted, with axis labels taken from the column names of data.

Value

No return value, called for side effects (produces a visualization of hard vs. soft cluster assignments from boot.kmeans results).

Author(s)

Jesse S. Ghashti <jesse.ghashti@ubc.ca> and Jeffrey L. Andrews <jeff.andrews@ubc.ca>

See Also

boot.kmeans, compare.clusters, bootk.hardsoftvis, kmeans, FKM

Examples

```
set.seed(1)
x <- as.matrix(iris[, -5])

# run bootstrap kmeans
res <- boot.kmeans(data = x, groups = 3, iterations = 20)

# scatterplot matrix of all variables
bootk.hardsoftvis(x, res, TRUE)

# scatterplot matrix of variable 1 and variable 2
bootk.hardsoftvis(x, res, plotallvars = FALSE, var1 = 1, var2 = 2)
```

`compare.clusters` *Compare traditional k-means, bootstrap augmented k-means, and fuzzy c-means*

Description

Fits three clustering procedures on the same data: standard `kmeans`, our bootstrap augmented k-means algorithm `boot.kmeans`, and (optionally) fuzzy c-means from `FKM`. Returns the fitted objects of all three whose object can be passed into `compare.clusters` to compare side-by-side confusion matrices.

Usage

```
compare.clusters( data = NULL,
                  groups = NULL,
                  seed = 13462,
                  nstart = 50,
                  what = "all")
```

Arguments

`data`	Numeric matrix or data frame of row observations and column variables. Required.
`groups`	Number of clusters K. Required.
`seed`	Optional integer random seed for reproducibility.
`nstart`	Number of random starts for initialization for all methods.
`what`	Character flag; if `"all"` (default), include fuzzy c-means (`FKM`) in the output.

Details

The function runs the following algorithms:

- `km`: `stats::kmeans(data, centers = groups, nstart = nstart)`.
- `bkm`: `boot.kmeans(data, groups, nstart = nstart, returnall = FALSE)`.
- `fkm` (if `what == "all"`): `fclust::FKM(data, k = groups, RS = nstart)`.

Value

A named list with components:

`km`	kmeans fit object.
`bkm`	`"BSKMeans"` object returned by `boot.kmeans`.
`fkm`	(Only if `what == "all"`) `fclust` fuzzy c-means fit.
`what`	Echo of the `what` argument.

References

Ghashti, J.S., Andrews, J.L., Thompson, J.R.J., Epp, J. and H.S. Kochar (2025). A bootstrap augmented k-means algorithm for fuzzy partitions. Submitted.

Bezdek, J.C. (1981). *Pattern recognition with fuzzy objective function algorithms*. New York: Plenum.

Hartigan, J.A. and M.A. Wong (1979). Algorithm AS 136: A K-means clustering algorithm. *Applied Statistics, 28*, 100–108.

Ferraro, M.B., Giordani P. and A. Serafini (2019). fclust: An R Package for Fuzzy Clustering, *The R Journal, 11*.

See Also

boot.kmeans, compare.tables, bootk.hardsoftvis, kmeans, FKM

Examples

```
set.seed(1)
x <- as.matrix(iris[, -5])

# compare all three methods
res <- compare.clusters(x, groups = 3, nstart = 10, what = "all")

# hard clusters from bootstrap kmeans
table(res$bkm$clusters, iris$Species)

# fuzzy memberships from fuzzy \eqn{c}-means
head(res$fkm$U)

# compare class labels
cbind(res$bkm$clusters[1:5], res$fkm$clus[1:5,2], res$km$cluster[1:5])
```

---

compare.tables *Contingency tables comparing true labels to fitted clusterings*

Description

Given the output of compare.clusters and a vector of true class labels, prints confusion tables for: (i) hard k-means labels, (ii) the bootstrap augmented k-means MAP out-of-bag labels, and (optionally) (iii) fuzzy c-means hard labels.

Usage

```
compare.tables(full.res = NULL, true.labs = NULL, verbose = TRUE)
```

Arguments

full.res	A list returned by compare.clusters, containing components km, bkm, and fkm (the latter only if argument what = "all" in function compare.clusters).
true.labs	A vector of true class labels.
verbose	Logical; if TRUE, prints the contingency tables to the console. Default is TRUE.

Details

For k-means, hard labels are taken from full.reskmcluster. For bootstrap k-means, labels are taken from full.resbkmclusters. If full.res$what == "all" results are also taken from full.resfkmclus, which are the hard cluster assignments from the fuzzy c-means algorithm.

The function prints two or three contingency tables to the console, with three presented if compare.clusters has argument what = "all", and two otherwise.

Value

A list with components:

kmeans	A contingency table comparing true labels to k-means cluster assignments.
bootkmeans	A contingency table comparing true labels to boot$k$means cluster assignments.
fuzzcmeans	(Optional) A contingency table comparing true labels to fuzzy c-means cluster assignments, included only if full.res$what == "all".

If verbose = TRUE, the tables are also printed to the console.

References

Ghashti, J.S., Andrews, J.L., Thompson, J.R.J., Epp, J. and H.S. Kochar (2025). A bootstrap augmented k-means algorithm for fuzzy partitions. Submitted.

Bezdek, J.C. (1981). *Pattern recognition with fuzzy objective function algorithms*. New York: Plenum.

Hartigan, J.A. and M.A. Wong (1979). Algorithm AS 136: A K-means clustering algorithm. *Applied Statistics, 28*, 100–108.

Ferraro, M.B., Giordani P. and A. Serafini (2019). fclust: An R Package for Fuzzy Clustering, *The R Journal, 11*.

See Also

boot.kmeans, compare.clusters, bootk.hardsoftvis, kmeans, FKM

Examples

```
set.seed(1)
x <- as.matrix(iris[, -5])

# fit three methods (kmeans, bootstrap kmeans, fuzzy \eqn{c}-means)
res <- compare.clusters(x, groups = 3, nstart = 10, what = "all")
```

```
# compare contigency tables
compare.tables(res, true.labs = iris$Species)
```

`fari` *Frobenius Adjusted Rand Index for Comparing Two Partition Matrices*

Description

Computes fuzzy generalizations of the Adjusted Rand Index based on Frobenius inner products of membership matrices. These measures extends the Adjusted Rand Index to compare fuzzy partitions.

Usage

```
fari(a, b)
```

Arguments

`a`	An $n \times G_1$ matrix of hard or fuzzy cluster memberships, where each row sums to 1.
`b`	An $n \times G_2$ matrix of hard or fuzzy cluster memberships, where each row sums to 1.

Value

A single numeric value

`fari`	The Frobenius Adjusted Rand index between `a` and `b`.

References

Andrews, J.L., Browne, R. and C.D. Hvingelby (2022). On Assessments of Agreement Between Fuzzy Partitions. *Journal of Classification, 39*, 326–342.

J.L. Andrews, FARI (2013). GitHub repository, https://github.com/its-likeli-jeff/FARI

Examples

```
set.seed(1)
a <- matrix(runif(600), nrow = 200, ncol = 3)
a <- a / rowSums(a)
b <- matrix(runif(600), nrow = 200, ncol = 3)
b <- b / rowSums(b)

fari(a, b)
```

Index