

# Package ‘cmbClust’

February 12, 2026

**Version** 0.0.2

**Date** 2026-02-11

**Title** Conditional Mixture Modeling and Model-Based Clustering

**Description** Conditional mixture model fitted via EM (Expectation Maximization) algorithm for model-based clustering, including parsimonious procedure, optimal conditional order exploration, and visualization.

**Depends** R ( $\geq 3.5.0$ )

**Imports** stats, parallel

**Suggests** HDclassif, MixSim, mclust

**License** GPL ( $\geq 2$ )

**Repository** CRAN

**LazyLoad** yes

**LazyData** no

**NeedsCompilation** yes

**Author** Yang Wang [aut, cre],  
Volodymyr Melnykov [aut],  
Stephen Moshier [ctb] (eigenvalue calculations in c),  
Rouben Rostamian [ctb] (memory allocation in c)

**Maintainer** Yang Wang <wangy00808@gmail.com>

**Date/Publication** 2026-02-12 07:10:08 UTC

## Contents

cmbClust-package . . . . .	2
ais . . . . .	2
cmb.em . . . . .	3
cmb.plot . . . . .	5
cmb.search . . . . .	6
smltn . . . . .	8

<b>Index</b>	<b>10</b>
--------------	-----------

---

cmbClust-package

*Conditional mixture modeling and model-based clustering*

---

### Description

The utility of this package includes fitting a conditional mixture model with EM (Expectation Maximization) algorithm, model-based clustering based on conditional mixture modeling, conditional mixture modeling with parsimonious procedures, and optimal conditional order exploration by using either a full search or the proposed searching algorithm, and illustration of clustering results through pairwise plots.

### Details

Function 'cmb.em' runs the parsimonious conditional mixture modeling for a user-specified conditioning order.

Function 'cmb.search' runs the 'cmb.em' procedure for all possible conditioning orders and determines the optimal order using BIC, or runs the proposed optimal order search algorithm and then the 'cmb.em' for the obtained optimal order.

Function 'cmb.plot' builds pairwise plots to present clustering results from functions 'cmb.em' and 'cmb.search'.

### Author(s)

Yang Wang and Volodymyr Melnykov.

Maintainer: Yang Wang <wangy4@cofc.edu>

### References

Melnykov, V., and Wang, Y. (2023). Conditional mixture modeling and model-based clustering. *Pattern Recognition*, 133, p. 108994.

---

ais

*Australian Institute of Sport data*

---

### Description

The data set considers body characteristics from 102 male and 100 female athletes at the Australian Institute of Sport. It is collected for a study of how data on various features varied with sport body size and sex of athlete.

### Usage

```
data(ais)
```

**Format**

A data frame with 202 observations on the following 13 variables.

**sex** Factor with levels: female, male;

**sport** Factor with levels: B\_Ball, Field, Gym, Netball, Row Swim, T\_400m, Tennis, T\_Sprnt, W\_Polo;

**RCC** Red cell count;

**WCC** White cell count;

**Hc** Hematocrit;

**Hg** Hemoglobin;

**Fe** Plasma ferritin concentration;

**BMI** Body Mass Index;

**SSF** Sum of skin folds;

**Bfat** Body fat percentage;

**LBM** Lean body mass;

**Ht** Height, cm;

**Wt** Weight, kg

**Details**

The data have been made publicly available in connection with the book by Cook, R.D. and Weisberg, S. (1994, ISBN-10:0471008397).

**References**

Cook, R.D. and Weisberg, S. (1994). *An introduction to regression graphics*. John Wiley & Sons.

---

cmb.em

*Conditional mixture modeling by EM algorithm*

---

**Description**

Runs conditional mixture modeling and model-based clustering by EM algorithm (Expectation Maximization) for a prespecified variables conditioning order. Runs variable selection procedure (forward, backward or stepwise) to achieve a parsimonious mixture model.

**Usage**

```
cmb.em(x, order = NULL, l, K, method = "stepwise", id0 = NULL, n.em = 200, em.iter = 5, EM.iter = 200, nk.min = NULL, max.spur=5, tol = 1e-06, silent = FALSE, Parallel = FALSE, n.cores = 4)
```

**Arguments**

x	dataset matrix (n x p)
order	customized variables' conditioning order (length p)
l	order of polynomial regression model
K	number of clusters
method	variable selection method (options 'stepwise', 'forward', 'backward' and 'none')
id0	initial membership vector (length n)
n.em	number of short EM in an emEM procedure
em.iter	maximum number of iterations of short EM in an emEM procedure
EM.iter	maximum number of EM iterations
nk.min	spurious output control
max.spur	number of trials
tol	tolerance level
silent	output control (TRUE/FALSE)
Parallel	parallel computing (TRUE/FALSE)
n.cores	number of cores in parallel computing

**Details**

In conditional mixture modeling, each component is modeled by a product of conditional distributions with the means expressed by polynomial regression functions depending on other variables. Polynomial regression function order  $l$  and the number of clusters  $K$  are prespecified by user. The model's initialization can be determined by passing a group membership vector to the argument `id`, or obtained by the emEM algorithm (the default setting) in the function. There are two arguments related to the emEM procedure, the number of short EM `n.em` and maximum number of iterations for short EM `em.iter`. By default, the `n.em = 200` and `em.iter = 5`. The method of variable selection can be specified as `method = "stepwise"`, `"forward"`, `"backward"`, or `"none"` where `method = none` means no parsimonious procedure conducted. During the model fitting and variable selection phases, EM algorithm will be applied multiple times, where options `EM.iter` and `tol` are stopping criteria of EM iteration. The spurious output control argument `nk.min`, by default `nk.min = (1 × (p - 1) + 1) × 2`, can be set by user. When spurious output is obtained, `cmb.em` will be rerun. The maximum number of rerunning is `max.spur`.

Notation:  $n$  - sample size,  $l$  - order of polynomial regression model,  $K$  - number of mixture components.

**Value**

data	input dataset
model	estimated regression models for each cluster ( $K \times p$ matrix)
id	vector of estimated membership (length n)
loglik	estimated log likelihood
BIC	Bayesian Information Criterion

Pi	vector of estimated mixing proportions (length K)
tau	matrix of estimated posterior probabilities (n x K)
beta	matrix of estimated regression parameters (K x (p + p(p-1)/2) )
s2	matrix of estimated variance (K x p)
order	applied conditioning order (length p)
n_pars	number of model parameters

## References

Biernacki C., Celeux G., Govaert G. (2003). Choosing Starting Values for the EM Algorithm for Getting the Highest Likelihood in Multivariate Gaussian Mixture Models. *Computational Statistics and Data Analysis*, 41(3-4), pp. 561-575.

## Examples

```
set.seed(1)
K <- 3
l <- 2
x <- as.matrix(iris[,-5])
id.true <- iris[,5]

# Run EM algorithm for fitting a conditioning mixture model
obj <- cmb.em(x = x, order = c(1,3,2,4), l, K, method = "stepwise", silent = FALSE,
Parallel = FALSE)
id.cmb <- obj$id
table(id.true, id.cmb)
obj$BIC
```

---

cmb.plot	<i>Graphic display for the results of conditional mixture modeling and model-based clustering</i>
----------	---

---

## Description

cmb.plot demonstrates the clustering results of functions cmb.em and cmb.search. A graph with a combination of pairwise scatter plot for data points, pairwise contour plot of estimated mixture density, and pairwise regression curves is produced.

## Usage

```
cmb.plot(obj, allcolors = NULL, allpch = NULL, lwd = 1, cex.text = 1, cex.point = 0.6,
mar = c(0.6,0.6,0.6,0.6), oma = c(3.5,3.5,2.5,14), nlevels = 30)
```

**Arguments**

obj	output object of the function <code>cmb.em()</code> or <code>cmb.search()</code>
allcolors	colours of clusters (length K)
allpch	styles of data points in clusters (length K)
lwd	line width, a positive number, defaulting to 1
cex.text	magnification of labels and titles, defaulting to 1
cex.point	magnification of plotting symbols, defaulting to 0.6
mar	margin sizes of plots in lines of text (length 4)
oma	outer margin sizes of a pairwise plot in lines of text (length 4)
nlevels	number of contour levels, defaulting to 30

**Value**

This function generates a graphic.

**Examples**

```
set.seed(4)
K <- 3
l <- 2
x <- as.matrix(iris[,-5])

# Run EM algorithm for fitting a conditioning mixture model

obj <- cmb.em(x = x, order = c(1,2,3,4), l, K, method = "stepwise",
             silent = TRUE, Parallel = FALSE)
cmb.plot(obj)
```

---

cmb.search

*Optimal conditioning order search*

---

**Description**

Runs forward, backward, or stepwise variable selection procedure for obtaining the parsimonious conditional mixture models when all conditional orders are considered. Alternatively, runs the optimal order search algorithm, and parsimonious conditional mixture modeling for the obtained order.

**Usage**

```
cmb.search(x, l, K, method = "stepwise", all.perms = TRUE, id0 = NULL, n.em = 200,
           em.iter = 5, EM.iter = 200, nk.min = NULL, max.spur = 5, tol = 1e-06, silent = FALSE,
           Parallel = TRUE, n.cores = 4)
```

**Arguments**

<code>x</code>	dataset matrix (n x p)
<code>l</code>	order of polynomial regression model
<code>K</code>	number of clusters
<code>method</code>	variable selection method (options 'stepwise', 'forward', 'backward' and 'none')
<code>all.perms</code>	conditioning order search algorithm (TRUE: full search; FALSE proposed search algorithm)
<code>id0</code>	initial group membership (length n)
<code>n.em</code>	number of short EM in emEM procedure
<code>em.iter</code>	maximum number of short EM iterations in emEM
<code>EM.iter</code>	maximum number of EM iterations
<code>nk.min</code>	spurious output control
<code>max.spur</code>	number of trials
<code>tol</code>	tolerance level
<code>silent</code>	output control
<code>Parallel</code>	Parallel computing
<code>n.cores</code>	number of cores in parallel computing

**Details**

Functions 'cmb.search' and 'cmb.em' have common arguments except 'all.perm'. With `all.perms = TRUE`, a full search is applied to data, that is running parsimonious conditional mixture modeling for all orders and recognizing the optimal order based on the BIC. Then two lists are returned: `best.model` stores the results for the conditional mixture model with the optimal order, and `models` has results for all orders. With the option `all.perms = FALSE`, the optimal conditional order search algorithm is applied, and then only the list `best.model` is returned.

**Value**

The list `models` is returned when `all.perms = TRUE`.

<code>best.model</code>	membership assignments and estimated parameters of mixture model with the optimal contioning order.
<b>data</b>	the input dataset
<b>model</b>	estimated regression models for each cluster (K x p)
<b>id</b>	vector of estimated membership (length n)
<b>loglik</b>	estimated log likelihood
<b>BIC</b>	Bayesian Information Criterion
<b>Pi</b>	vector of estimated mixing proportions (length K)
<b>tau</b>	matrix of estimated posterior probabilities (n x K)
<b>beta</b>	matrix of estimated regression parameters (K x (p + p(p-1)/2) )
<b>s2</b>	matrix of estimated variances (K x p)
<b>order</b>	applied conditioning order (length p)

**n\_pars** number of parameters

models membership assignments and model parameters of mixture models with all conditioning orders.

**model** list of estimated regression models for all clusters ( $K \times p \times p!$ )

**id**  $p!$  vectors of estimated memberships ( $n \times p!$ )

**loglik** estimated log likelihood values (length  $p!$ )

**BIC** Bayesian Information Criterion values (length  $p!$ )

**Pi**  $p!$  vectors of estimated mixing proportions ( $K \times p!$ )

**tau**  $p!$  matrices of estimated posterior probabilities ( $K \times p!$ )

**beta**  $p!$  matrices of estimated regression parameters ( $K \times (p + p(p-1)/2) \times p!$ )

**s2**  $p!$  matrices of estimated variances ( $K \times p \times p!$ )

**order** applied conditioning orders ( $p! \times p$ )

**n\_pars** number of parameters in  $p!$  models (length  $p!$ )

### See Also

cmb.em

### Examples

```
set.seed(1)
K = 3
l <- 2
x <- as.matrix(iris[,-5])

obj <- cmb.search(x = x, l, K, method = "stepwise", all.perms = FALSE,
Parallel = FALSE, silent = FALSE)
obj$best.model$BIC
```

---

smltn

*Simulated datasets*

---

### Description

Datasets are simulated from conditional mixture models with different numbers of components.

### Usage

```
data(smltn)
```

### Format

Two datasets are stored in the data `smltn`. `smltn1` is a data matrix with 200 observations on two variables and one group membership; `smltn2` is a matrix with 300 observations on two variables and one group ID.



**Examples**

```
data(smltn)
# view data matrices smltn1 and smltn2
print(smltn1)
print(smltn2)
```

# Index

- \* **EM algorithm**
    - cmb.em, 3
  - \* **Model-based clustering**
    - cmb.em, 3
  - \* **Optimal conditioning order search**
    - cmb.search, 6
  - \* **Pairwise plot**
    - cmb.plot, 5
  - \* **conditional mixture modeling**
    - cmb.em, 3
  - \* **datasets**
    - ais, 2
    - smltn, 8
  - \* **variable selection**
    - cmb.em, 3
- ais, 2
- cmb.em, 3  
cmb.plot, 5  
cmb.search, 6  
cmbClust-package, 2
- smltn, 8  
smltn1 (smltn), 8  
smltn2 (smltn), 8