

# Package ‘sensory’

July 23, 2025

**Type** Package

**Title** Simultaneous Model-Based Clustering and Imputation via a  
Progressive Expectation-Maximization Algorithm

**Version** 1.1

**Date** 2016-02-23

**Author** Brian C. Franczak, Ryan P. Browne and Paul D. McNicholas

**Maintainer** Brian C. Franczak <bfrancza@math.mcmaster.ca>

**Description** Contains the function CUUimpute() which performs model-  
based clustering and imputation simultaneously.

**Depends** Matrix, gtools, MASS, R (>= 3.2.2)

**NeedsCompilation** no

**License** GPL (>= 2)

**Repository** CRAN

**Date/Publication** 2016-02-23 23:02:20

## Contents

|   |          |
|---|----------|
| CUUimpute . . . . .                         | 2        |
| sensory . . . . .                           | 4        |
| The Brown Bread Data . . . . .              | 5        |
| The Iris Data with Missing Values . . . . . | 5        |
| The White Bread Data . . . . .              | 6        |
| <b>Index</b>                                | <b>7</b> |

CUUimpute

*Cluster-Wise Imputation via a PEM Algorithm***Description**

Simultaneously performs model-based clustering and imputation using a Parsimonious Gaussian Mixture Model (PGMM) with CUU covariance structure.

**Usage**

```
CUUimpute(x, G = 1:3, q = 1:2, epsilon = 0.01, max.iter = 10000, known = NULL,
print = TRUE)
```

**Arguments**

|          |   |
|----------|---|
| x        | A numeric matrix.   |
| G        | A number or vector indicating the number components to fit.   |
| q        | A number or vector indicating the number of latent factors to fit.  |
| epsilon  | The tolerance value for Aitken's acceleration.  |
| max.iter | The maximum number of iterations for the PEM algorithm. A warning message is displayed if the maximum is met.                                 |
| known    | Optional. A vector of group memberships that must be numeric and whose length must be equal to the number of rows in x. (See Example I below) |
| print    | Logical indicating whether or not the iteration number and the corresponding log-likelihood value should be printed.                          |

**Details**

The PGMM with CUU covariance structure, herein referred to as the CUU model, was developed in McNicholas and Murphy (2008) for model-based clustering. The CUU model, like the other PGMMs, arises by assuming a latent Gaussian model structure for each population. As a result, the number of free parameters in CUU model's covariance structure increases linearly as the dimension of the data increases.

Browne et al. (2013) developed a Progressive Expectation-Maximization (PEM) algorithm to fit the CUU model to data with missing values. Under this parameter estimation scheme the CUU model is able to simultaneously impute and cluster a given data matrix. The CUUimpute() function fits the CUU model for a varying number of components, G, and latent factors, q, to a data matrix x. The function will run if only a data matrix is supplied however, the user is able to choose how many components and latent factors to fit, specify the tolerance limit for Aitken's acceleration, the maximum number of PEM iterations the algorithm will perform and give a vector representing the group memberships of each observation, if that information exists.

**Value**

|                          |  |
|--------------------------|--|
| <code>allbic</code>      | An array containing the Bayesian Information Criterion (BIC) values for each CUU model fitted to the data.   |
| <code>bic</code>         | The BIC of the best fitting CUU model.   |
| <code>G</code>           | The number of components in the best fitting CUU model.  |
| <code>q</code>           | The number of latent factors in the best fitting CUU model.  |
| <code>loglik</code>      | A vector whose length is equal to the number of PEM iterations performed and whose elements contain the log-likelihood value on each iteration.  |
| <code>gpar</code>        | A list of the model parameters corresponding to the best fitting CUU model.  |
| <code>yhat</code>        | A data matrix with imputed values.   |
| <code>u</code>           | A matrix containing the values of the latent variables.  |
| <code>zig</code>         | A matrix giving the probabilities of group membership for each observation. Note: these probabilities are based off the model parameters of the best fitting CUU model   |
| <code>map</code>         | A vector whose length is equal to the number of observations and whose elements correspond to the group membership of each observation.  |
| <code>class.table</code> | A cross tabulation between the predicted and true group memberships.   |
| <code>iclresult</code>   | A list containing all of the information listed above for the best fitting CUU model chosen by the Integrated Complete Likelihood (ICL) measure. Note: <code>allbic</code> and <code>bic</code> are replaced by <code>allicl</code> and <code>icl</code> . |

**Author(s)**

Brian C. Franczak, Ryan P. Browne and Paul D. McNicholas  
 Maintainer: Brian C. Franczak <bfranca@math.mcmaster.ca>

**References**

P.D. McNicholas and T.B. Murphy (2008). Parsimonious Gaussian Mixture Models. *Statistics and Computing* (18), 285-296.  
 Browne, R.P., P.D. McNicholas, and C.J. Findlay (2013). A partial EM algorithm for clustering white breads. *arXiv preprint arXiv:1302.6625*.

**See Also**

[sensory](#)

**Examples**

```
### Example I
data(iiris) # loads the modified Iris data
head(iiris) # displays the first six rows of the modified Iris data
data(iris) # loads the original Iris data

# create a vectors whose elements contain the group memberships of each flower
iris.known <- as.integer(iris[,ncol(iris)])
```

```
# fit a three component CUU models to the modified Iris data
output <- CUUimpute(x=iiris,G=3,q=1,print=FALSE,known=iris.known)
output
summary(output) # Summarizes the results

names(output) # Shows what results are available
output$allbic # Gives every BIC value
output$bic # Gives BIC for best fitting model
output$G # Gives the number of components for the best fitting CUU model
output$loglik # Gives a vector whose elements are the log-likelihood at each iteration
output$gpar # Gives the model parameters for the best fitting CUU model
head(output$yhat) # Displays the first 6 rows of the imputed matrix
output$yhat # Gives the entire imputed matrix
output$class.table # Gives a classification table between the predicted and true group memberships
```

---

sensory

---

*Cluster-Wise Imputation via a PEM Algorithm*


---

## Description

Contains the function `CUUimpute()` which performs model-based clustering and imputation simultaneously.

## Details

|          |            |
|----------|------------|
| Package: | sensory    |
| Type:    | Package    |
| Version: | 1.0        |
| Date:    | 2014-08-07 |
| License: | GPL (>=2)  |

This package contains two data sets collected at Compusense Inc. in Guelph, Ontario, Canada, a modified version of Fisher's Irises, and the function `CUUimpute()`.

## Author(s)

Brian C. Franczak, Ryan P. Browne and Paul D. McNicholas  
 Maintainer: Brian C. Franczak <bfranca@math.mcmaster.ca>

## References

P.D. McNicholas and T.B. Murphy (2008). Parsimonious Gaussian Mixture Models. *Statistics and Computing* (18), 285-296.  
 Browne, R.P., P.D. McNicholas, and C.J. Findlay (2013). A partial EM algorithm for clustering white breads. *arXiv preprint arXiv:1302.6625*.

**See Also**

Details, examples, and references are given under [CUUimpute](#)

---

The Brown Bread Data    *The Brown Bread Data Set*

---

**Description**

The brown bread data set is composed of 570 panelists and 16 products, labelled A,...,P to protect the identity of the manufacturer. Each panelist rated a subset of 6 products using the 9-point Hedonic Scale.

**Usage**

```
data(bbread)
```

**Examples**

```
data(bbread) # Loads the brown bread data set
head(bbread) # Displays the first six rows of the brown bread data set
```

---

The Iris Data with Missing Values  
*The Iris Data with Missing Values*

---

**Description**

Fisher's Irises with one measurement removed per observation. For details load `data(iris,package="gclus")`.

**Usage**

```
data(iiris)
```

**Examples**

```
data(iiris) # Loads the modified Iris data.
head(iiris) # Displays the first six rows of the modified Iris Data
```

---

The White Bread Data    *The White Bread Data*

---

**Description**

The white bread data is a liking study composed of 420 panelists and 12 products, labelled A,...,L to protect the identity of the manufacturer. Each panelist rated 6 white breads using the 9-point Hedonic scale.

**Usage**

```
data(wbread)
```

**References**

Browne, R.P., P.D. McNicholas, and C.J. Findlay (2013). A partial EM algorithm for clustering white breads. arXiv preprint arXiv:1302.6625.

**Examples**

```
data(wbread) # Loads the white bread data  
head(wbread) # Displays the first six rows of the white bread data
```

# Index

- \* **Brown Bread Data**
  - The Brown Bread Data, [5](#)
- \* **CUU Model**
  - CUUimpute, [2](#)
  - sensory, [4](#)
- \* **Clustering**
  - CUUimpute, [2](#)
  - sensory, [4](#)
- \* **Expectation Maximization (EM)**
  - Algorithm**
    - CUUimpute, [2](#)
    - sensory, [4](#)
- \* **Fisher's Irises**
  - The Iris Data with Missing Values, [5](#)
- \* **Imputation**
  - CUUimpute, [2](#)
  - sensory, [4](#)
- \* **Parsimonious Gaussian Mixture Models**
  - CUUimpute, [2](#)
  - sensory, [4](#)
- \* **Sensory Informed Designs**
  - CUUimpute, [2](#)
  - sensory, [4](#)
  - The Brown Bread Data, [5](#)
  - The White Bread Data, [6](#)
- \* **White Bread Data**
  - The White Bread Data, [6](#)
- \* **datasets**
  - The Brown Bread Data, [5](#)
  - The Iris Data with Missing Values, [5](#)
  - The White Bread Data, [6](#)
- 
- bbread(The Brown Bread Data), [5](#)
- 
- CUUimpute, [2](#), [5](#)
- 
- iiris(The Iris Data with Missing Values), [5](#)
- 
- sensory, [3](#), [4](#)
- 
- The Brown Bread Data, [5](#)
- The Iris Data with Missing Values, [5](#)
- The White Bread Data, [6](#)
- 
- wbread(The White Bread Data), [6](#)