

# Package: fakmct (via r-universe)

August 28, 2024

**Type** Package

**Title** Fuzzy Adaptive Resonance Theory K-Means Clustering Technique

**Version** 0.1.0

**Author** Alfi Nurrahmah [aut, cre] Budi Yuniarto [aut]

**Maintainer** Alfi Nurrahmah <221810140@stis.ac.id>

**Description** A set of function for clustering data observation with hybrid method Fuzzy ART and K-Means by Sengupta, Ghosh & Dan (2011) <[doi:10.1080/0951192X.2011.602362](https://doi.org/10.1080/0951192X.2011.602362)>.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.2

**URL** <<https://github.com/alfinurrahmah/fakmct>>

**Imports** stats

**Depends** R (>= 3.5.0)

**Repository** <https://alfinurrahmah.r-universe.dev>

**RemoteUrl** <https://github.com/alfinurrahmah/fakmct>

**RemoteRef** HEAD

**RemoteSha** 60cefd346378e97921c5315275e45d45bb415034

## Contents

choice_function . . . . .	2
fakmct . . . . .	2
fuzzy_and . . . . .	4
fuzzy_norm . . . . .	5
linalg_norm . . . . .	5
match_function . . . . .	6
simulatedataIPM . . . . .	6
update_weight . . . . .	7
vigilance_check . . . . .	7

**Index****8**


---

choice_function	<i>Choice Function</i>
-----------------	------------------------

---

**Description**

Calculates the similarity between the input pattern I and all of saved categories.

**Usage**

```
choice_function(input, category_w, alpha)
```

**Arguments**

input	The input (vector) data observation
category_w	The current category weight
alpha	Choice parameter $\alpha > 0$

**Value**

Returns the vector of Tj choice activation function

---

fakmct	<i>Fuzzy Adaptive Resonance Theory (ART) K-Means Clustering Technique</i>
--------	---

---

**Description**

Clustering data observation with hybrid method Fuzzy ART and K-Means

**Usage**

```
fakmct(
  input,
  rho,
  alpha,
  beta,
  w_init = NA,
  max_epochs = 1000,
  max_clusters = 1000,
  eps = 10^-6
)
```

**Arguments**

input	The input (vector) data observation. Should be numeric type of data.
rho	Vigilance parameter in (0,1)
alpha	Choice parameter alpha > 0
beta	Learning rate in (0,1)
w_init	Initial weight
max_epochs	Maximum number of iterations
max_clusters	Maximum number of clusters that allowed
eps	Tolerance with default is $10^{-6}$

**Value**

labels	clusters label of each observations
size	the size of each clusters that have been formed
clusters	a list of observations in each clusters
centroids	cluster centroids that are calculated by the mean value of the objects in each clusters
weights	the model weight
params	parameters that have been saved in the function
num_clusters	number of cluster that have been formed
running.time	time for running function

**Examples**

```

library(fakmct)
# Using dataset iris
## load data
data.inputs = iris[,-5]
true.labels = as.numeric(unlist(iris$Species))

## run model data
ex.iris<-fakmct(data.inputs, alpha = 0.3, rho = 0.5, beta = 1, max_epochs = 50, max_clusters = 5)
ex.iris$labels
ex.iris$size
ex.iris$centroids
ex.iris$params

## plot data
plot(data.inputs, col = ex.iris$labels, pch = true.labels,
      main = paste0("Dataset: Iris"))

# Using data IPM 2019

## load simulate data IPM
data("simulatedataIPM")
dt <- simulatedataIPM

```

```
## run model data IPM
mod.fakm<-fakmct(dt, alpha = 0.3, rho = 0.5, beta = 0.1, max_epochs = 50, max_clusters = 5)
mod.fakm$labels
mod.fakm$size
mod.fakm$centroids
mod.fakm$params

## plot data IPM
plot(dt, col = mod.fakm$labels, pch=mod.fakm$labels, main = paste0("Dataset IPM"))
```

---

fuzzy\_and

*Fuzzy And Function*

---

## Description

Fuzzy And Function

## Usage

```
fuzzy_and(inputA, inputB)
```

## Arguments

inputA	First input vector
inputB	Second input vector. Must be of the same dimension as inputA.

## Value

Returns the Fuzzy AND of two input values in a vector.

## Examples

```
fuzzy_and(0, -1) # = -1
fuzzy_and(0, 1) # = 0
fuzzy_and(1, 2) # = 1
fuzzy_and(1, 1) # = 1
fuzzy_and(c(0.5, 0.75), c(1.5, 1)) # = c(0.5, 0.75)
```

---

`fuzzy_norm`*Fuzzy Norm*

---

**Description**

Fuzzy Norm

**Usage**`fuzzy_norm(input)`**Arguments**`input`            The input (vector) data observation**Value**

Returns the Fuzzy norm results of input values

**Examples**

```
a = c(-1,-3,4,5)
fuzzy_norm(a) # = 13
```

---

`linalg_norm`*Linear Algebra for Euclidean distance*

---

**Description**

Linear Algebra for Euclidean distance

**Usage**`linalg_norm(inputA, inputB)`**Arguments**`inputA`            First input vector  
`inputB`            Second input vector. Must be of the same dimension as inputA.**Value**

Returns the calculation results by squares of distances between two input values

**Examples**

```
a <- c(-3,-2,-1,3,3,2,3)
b <- c(-3,-2,-1,0,1,2,3)
linalg_norm(a,b) # = 3.605
```

---

match_function	<i>Match function</i>
----------------	-----------------------

---

**Description**

Match function

**Usage**

```
match_function(input, category_w)
```

**Arguments**

input	The input (vector) data observation
category_w	The current category weight

**Value**

Returns the vector of match  $S_j$  that will be used to check the vigilance parameter

---

simulatedataIPM	<i>Sample Data for simulate analysis data (Using IPM 2019)</i>
-----------------	--

---

**Description**

A real data of Human Development Index (Indeks Pembangunan Manusia) of West Java, Indonesia 2019

**Usage**

```
simulatedataIPM
```

**Format**

A tibble with 27 observation as region and 4 column as variables, which are:

- "AHH" a value of Life expectancy (Angka Harapan Hidup)
- "HLS" a value of Expected Years of Schooling (Harapan Lama Sekolah)
- "RLS" a value of Mean Years of Schooling (Rata-rata Lama Sekolah)
- "Pengeluaran" a value of Expenditure (Pengeluaran)

**Source**

<https://www.bps.go.id/>

---

update_weight	<i>Update weight</i>
---------------	----------------------

---

**Description**

Update weight

**Usage**

```
update_weight(input, category_w, beta)
```

**Arguments**

input	The input (vector) data observation
category_w	The current category weight
beta	Learning rate in (0,1)

**Value**

Returns the updated weight

---

vigilance_check	<i>Vigilance check</i>
-----------------	------------------------

---

**Description**

Vigilance check

**Usage**

```
vigilance_check(input, category_w, rho)
```

**Arguments**

input	The input (vector) data observation
category_w	The current category weight
rho	Vigilance parameter (0,1)

**Value**

Returns Boolean value (True or False) as a result of checking the match  $S_j$  vector passed the vigilance parameter or not

# Index

## \* datasets

simulatedataIPM, 6

choice\_function, 2

fakmct, 2

fuzzy\_and, 4

fuzzy\_norm, 5

linalg\_norm, 5

match\_function, 6

simulatedataIPM, 6

update\_weight, 7

vigilance\_check, 7