Package ‘fcm’

October 13, 2022

Type Package

Title Inference of Fuzzy Cognitive Maps (FCMs)

Version 0.1.3

Date 2017-07-09

Author
Zoumpoulia Dikopoulou <dikopoulia@gmail.com> <zoumpolia.dikopoulou@uhasselt.be>, Elpiniki Papageorgiou <epapageorgiou@teiste.gr>, <e.i.papageorgiou75@gmail.com>

Maintainer Zoumpoulia Dikopoulou <dikopoulia@gmail.com>

Description Provides a selection of 3 different inference rules (including additionally the clamped types of the referred inference rules) and 4 threshold functions in order to obtain the inference of the FCM (Fuzzy Cognitive Map). Moreover, the 'fcm' package returns a data frame of the concepts' values of each state after the inference procedure. Fuzzy cognitive maps were introduced by Kosko (1986) <doi:10.1002/int.4550010405> providing ideal causal cognition tools for modeling and simulating dynamic systems.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Imports reshape2, ggplot2

VignetteBuilder knitr

Suggests knitr, rmarkdown

NeedsCompilation no

Repository CRAN

Date/Publication 2017-07-09 22:44:14 UTC

R topics documented:

fcm *Inference of Fuzzy Cognitive Maps (FCMs)*

Description

Estimates the Inference of a Fuzzy Cognitive Map. Provides a selection of 3 different inference rules (including 3 more, the clamped types of inference rules) and 4 threshold functions in order to obtain the FCM inference procedure. FCM (Kosko, 1986) is proven to be capable of causal inference and is applicable to complex decision problems where numerous interlinked dependent variables influence one another. The fcm package includes the fcm.infer function which returns a dataframe of the concepts' values of each state after the inference procedure.

fcm.infer *Inference of Fuzzy Cognitive Maps (FCMs)*

Description

Estimates the Inference of a Fuzzy Cognitive Map. Moreover, the 'fcm' package provides a selection of 3 different inference rules (and 3 more types derived from the clamped inference rules) and 4 threshold functions in order to obtain the FCM inference. FCM (Kosko, 1986) is proven to be capable of causal inference and is applicable to complex decision problems where numerous interlinked dependent variables influence one another.

Usage

```
fcm.infer(activation_vec, weight_mat, iter = 20, infer = "k",
  transform = "s", lambda = 1, e = 0.001)
```

Arguments

`activation_vec`	A [1 x m] data frame which contains the initial concept values (m). A concept is turned on or activated by making its vector element 1 or 0 or in [0, 1].
`weight_mat`	m x m data frame which stores the weights assigned to the pairs of concepts. The weights are usually normalized to the interval [0,1] or [-1, +1].
`iter`	The required number of iterations in order to reach the FCM convergence. Defaults to 20.
`infer`	Select an Inference Rule ('k' Kosko, 'mk' modified Kosko, 'r' Rescale,'kc' (and the clamped types of inference rules: Kosko-clamped, 'mkc' modified Kosko-clamped or 'rc' Rescale-clamped)). Default value is set to 'k'
`transform`	Contains the Transformation functions ('b' Bivalent, 'tr' Trivalent, 's' Sigmoid or 't' Hyperbolic tangent). The transformation function is used to reduce unbounded weighted sum to a certain range, which hinders quantitative analysis, but allows for qualitative comparisons between concepts. Default value is set equal to 's'.

`lambda` A parameter that determines the steepness of the sigmoid and hyperbolic tangent function at values around 0. Different lambda value may perform more appropriate for different problems.

`e` Epsilon (e) is a residual, describing the minimum error difference among the subsequent concepts. Its value depends on the application type. Defaults to 0.001.

Value

Returns [iter x m] data frame which contains the concepts' values of each iteration after the the transformation function.

Author(s)

Zoumpoulia Dikopoulou <dikopoulia@gmail.com>, <zoumpolia.dikopoulou@uhasselt.be>

Elpiniki Papageorgiou <epapageorgiou@teiste.gr>, <e.i.papageorgiou75@gmail.com>

References

B. Kosko, "Fuzzy cognitive maps", International Journal of Man-Machine Studies 24, p.p. 65-75, 1986.

C.D. Stylios, P.P. Groumpos, "A Soft Computing Approach for Modelling the Supervisor of Manufacturing Systems", Intelligent and Robotic Systems, vol. 26, p.p. 389-403, 1999.

E.I. Papageorgiou, "A new methodology for Decisions in Medical Informatics using fuzzy cognitive maps based on fuzzy rule-extraction techniques", Applied Soft Computing, vol. 11, Issue 1, p.p. 500-513, 2011.

E.I. Papageorgiou, "Fuzzy Cognitive Maps for Applied Sciences and Engineering From Fundamentals to Extensions and Learning Algorithms", Intelligent Systems Reference Library, Vol 54, 2014.

Examples

```
## Not run:
# Example for the FCM inference with 8 nodes


### Input data

# Create the activation vector
act.vec <- data.frame(1, 1, 0, 0, 0, 0, 0, 0)
# Change the column names
colnames(act.vec) <- c("C1", "C2", "C3", "C4", "C5", "C6", "C7", "C8")

C1 = c(0.0, 0.0, 0.6, 0.9, 0.0, 0.0, 0.0, 0.8)
C2 = c(0.1, 0.0, 0.0, 0.0, 0.0, 0.0, 0.2, 0.5)
C3 = c(0.0, 0.7, 0.0, 0.0, 0.9, 0.0, 0.4, 0.1)
C4 = c(0.4, 0.0, 0.0, 0.0, 0.0, 0.9, 0.0, 0.0)
C5 = c(0.0, 0.0, 0.0, 0.0, 0.0, -0.9, 0.0, 0.3)
C6 = c(-0.3, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
```

```
C7 = c(0.0, 0.0, 0.0, 0.0, 0.0, 0.8, 0.4, 0.9)
C8 = c(0.1, 0.0, 0.0, 0.0, 0.0, 0.1, 0.6, 0.0)

# Create the weight matrix
w.mat <- matrix(c(C1, C2, C3, C4, C5, C6, C7, C8),
                nrow = 8, ncol = 8, byrow = TRUE)
colnames(w.mat) <- c("C1", "C2", "C3", "C4", "C5", "C6", "C7", "C8")
w.mat <- as.data.frame(w.mat)



### Select the arguments for the fcm.infer function

output <- fcm.infer(act.vec, w.mat, 50, "r", "s")
View(output$values)   # View the concept values for each iteration


### Visualize the concepts' values for each state

library (reshape2)
library (ggplot2)
# create a numeric vector named "iterations"
iterations <- as.numeric(rownames(output$values))
# add "iterations" in the "output$values" dataframe
df <- data.frame(iterations, output$values)
#transform df into long formats
df2 <- melt(df, id="iterations")
ggplot(data=df2,
       aes(x=iterations, y=value, group=variable, colour=variable)) +
       theme_bw() + geom_line(size=0.7) + geom_point(size = 2)

## End(Not run)
```

Index