

# Package ‘FisPro’

June 14, 2019

**Type** Package

**Title** Fuzzy Inference System Design and Optimization

**Version** 1.0

**Author** Serge Guillaume [aut],  
Brigitte Charnomordic [aut],  
Jean-Luc Lablée [aut, cre],  
Hazaël Jones [ctb],  
Lydie Desperben [ctb],  
IRSTEA [cph] (Institut national de Recherche en Sciences et  
Technologies pour l'Environnement et l'Agriculture, France),  
INRA [cph] (Institut National de la Recherche Agronomique, France)

**Maintainer** Jean-Luc Lablée <jean-luc.lablee@irstea.fr>

**URL** <https://www.fispro.org>

**Description** Fuzzy inference systems are based on fuzzy rules, which have a good capability for managing progressive phenomenons.

This package is a basic implementation of the main functions to use a Fuzzy Inference System (FIS) provided by the open source software 'FisPro' <<https://www.fispro.org>>.

'FisPro' allows to create fuzzy inference systems and to use them for reasoning purposes, especially for simulating a physical or biological system.

**License** CeCILL

**Encoding** UTF-8

**Depends** R (>= 3.3.0)

**Imports** methods, Rdpack, Rcpp (>= 1.0.0)

**RdMacros** Rdpack

**LinkingTo** Rcpp

**NeedsCompilation** yes

**Suggests** testthat

**RoxygenNote** 6.1.1

**Repository** CRAN

**Date/Publication** 2019-06-14 10:40:08 UTC

## R topics documented:

FisPro-package . . . . .	2
fis . . . . .	3
mf . . . . .	4
mf_trapezoidal . . . . .	4
mf_trapezoidal_inf . . . . .	5
mf_trapezoidal_sup . . . . .	5
mf_triangular . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

FisPro-package	<i>FisPro package</i>
----------------	-----------------------

---

### Description

This package is a basic implementation of the main functions to use a Fuzzy Inference Systems that can be used for reasoning purposes, especially for simulating a physical or biological system. It is derived from the 'FisPro' open source software <https://www.fispro.org>. Fuzzy inference systems are briefly described in the [Fuzzy Logic Elementary Glossary](#). They are based on fuzzy rules, which have a good capability for managing progressive phenomenon. Fuzzy logic, since the pioneer work by Zadeh, has proven to be a powerful interface between symbolic and numerical spaces. One of the reasons for this success is the ability of fuzzy systems to incorporate human expert knowledge with its nuances, as well as to express the behaviour of the system in an interpretable way for humans. Another reason is the possibility of designing data-driven FIS to make the most of available data.

To design a fuzzy system that can be handled by this package the user can use the 'FisPro' software. If needed, the package can be extended to other functions.

Enjoy FisPro!

### Author(s)

FisPro Team <contact@fispro.org>

### References

<https://www.fispro.org>

Guillaume S, Charnomordic B (2011). "Learning interpretable Fuzzy Inference Systems with FisPro." *International Journal of Information Sciences*, **181**(20), 4409-4427. doi: [10.1016/j.ins.2011.03.025](https://doi.org/10.1016/j.ins.2011.03.025), Special Issue on Interpretable Fuzzy Systems.

Guillaume S, Charnomordic B (2012). "Fuzzy Inference Systems: an integrated modelling environment for collaboration between expert knowledge and data using FisPro." *Expert Systems with Applications*, **39**, 8744-8755. doi: [10.1016/j.eswa.2012.01.206](https://doi.org/10.1016/j.eswa.2012.01.206).

---

fis	<i>fis class</i>
-----	------------------

---

## Description

The `fis` class is the main class (Rcpp class) to open and use a Fuzzy Inference System.

## Fields

`name` the name of the FIS.

`input_size` the number of inputs in the FIS.

`output_size` the number of outputs in the FIS.

## Methods

**constructor** argument: `fis_file` a FIS configuration file (can be built with FisPro software <https://www.fispro.org>).  
return: an object of `fis`.

**infer** infers a value for each output from the input values  
argument: `values` numerical vector of input values.  
return: all inferred outputs.

**infer\_output** infers a value for a single output from the input values  
argument: `values` numerical vector of input values.  
argument: `output_number` the number of the output to infer.  
return: the inferred output.

## References

[Fuzzy Logic Elementary Glossary](#)

## Examples

```
# build the FIS
fis_file <- system.file("extdata", "test.fis", package = "FisPro")
fis <- new(fis, fis_file)

# infer all outputs
inferred <- fis$infer(c(0.25, 0.75))
#infer first output
inferred_output1 <- fis$infer_output(c(0.25, 0.75), 0)
#infer second output
inferred_output2 <- fis$infer_output(c(0.25, 0.75), 1)
```

---

mf	<i>mf class</i>
----	-----------------

---

### Description

Base class of all MF classes. This class cannot be instantiated. Use derived classes [mf\\_triangular](#), [mf\\_trapezoidal](#), [mf\\_trapezoidal\\_inf](#) or [mf\\_trapezoidal\\_sup](#)

### Fields

name the name of the mf.

### Methods

degree(value) compute the membership degree of a numerical value.  
 argument: value numerical value to compute the membership degree.  
 return: the membership degree.

### References

[Fuzzy Logic Elementary Glossary](#)

---

mf_trapezoidal	<i>mf_trapezoidal class</i>
----------------	-----------------------------

---

### Description

Class to build trapezoidal MF.

### Inherits

mf\_trapezoidal class inherits all fields and methods of [mf](#) class.

### Methods

constructor(lower\_support, lower\_kernel, upper\_kernel, upper\_support) argument: lower\_support numerical lower value of support.  
 argument: lower\_kernel numerical lower value of kernel.  
 argument: upper\_kernel numerical upper value of kernel.  
 argument: upper\_support numerical upper value of support.  
 return: an object of mf\_trapezoidal.

### Examples

```
mf <- new(mf_trapezoidal, 0, 1, 2, 3)
mf$degree(0.5)
```

---

mf\_trapezoidal\_inf      *mf\_trapezoidal\_inf class*

---

**Description**

Class to build trapezoidal inf MF.

**Inherits**

mf\_trapezoidal\_inf class inherits all fields and methods of [mf](#) class.

**Methods**

constructor(upper\_kernel, upper\_support) argument: upper\_kernel numerical upper value of kernel.  
argument: upper\_support numerical upper value of support.  
return: an object of mf\_trapezoidal\_inf.

**Examples**

```
mf <- new(mf_trapezoidal_inf, 0, 1)
mf$degree(0.5)
```

---

mf\_trapezoidal\_sup      *mf\_trapezoidal\_sup class*

---

**Description**

Class to build trapezoidal sup MF.

**Inherits**

mf\_trapezoidal\_sup class inherits all fields and methods of [mf](#) class.

**Methods**

constructor(lower\_support, lower\_kernel) argument: lower\_support numerical lower value of support.  
argument: lower\_kernel numerical lower value of kernel.  
return: an object of mf\_trapezoidal\_sup.

**Examples**

```
mf <- new(mf_trapezoidal_sup, 0, 1)
mf$degree(0.5)
```

---

mf_triangular	<i>mf_triangular class</i>
---------------	----------------------------

---

**Description**

Class to build triangular MF.

**Inherits**

mf\_triangular class inherits all fields and methods of [mf](#) class.

**Methods**

constructor(lower\_support, kernel, upper\_support) argument: lower\_support numerical lower value of support.  
argument: kernel numerical value of kernel.  
argument: upper\_support numerical upper value of support.  
return: an object of mf\_triangular.

**Examples**

```
mf <- new(mf_triangular, 0, 1, 2)
mf$degree(0.5)
```

# Index

`fis`, [3](#)

FisPro-package, [2](#)

`mf`, [4](#), [4](#), [5](#), [6](#)

`mf_trapezoidal`, [4](#), [4](#)

`mf_trapezoidal_inf`, [4](#), [5](#)

`mf_trapezoidal_sup`, [4](#), [5](#)

`mf_triangular`, [4](#), [6](#)