Package 'fugeR'

February 19, 2015

Maintainer Alexandre Bujard <alexandre.bujard@gmail.com>

License GPL (>= 2.0)

Title FUzzy GEnetic, a machine learning algorithm to construct prediction model based on fuzzy logic.

Type Package

LazyLoad yes

Author Alexandre Bujard <alexandre.bujard@gmail.com>

Description This is an evolutionary algorithm for fuzzy systems, a genetic algorithm is used to construct a fuzzy system able to fit the given training data. This fuzzy system can then be used as a prediction model, it's composed of fuzzy logic rules that provide a good lingustic representation.

Version 0.1.2

Date 2012-07-11

Depends R (>= 2.15.1), snowfall, Rcpp (>= 0.9.12)

LinkingTo Rcpp

Repository CRAN

Date/Publication 2012-08-08 13:29:39

NeedsCompilation yes

R topics documented:

fugeR-package *FUzzy GEnetic, a machine learning algorithm to construct prediction model based on fuzzy logic.*

Description

This is an evolutionary algorithm for fuzzy systems, a genetic algorithm is used to construct a fuzzy system able to fit the given training data. This fuzzy system can then be used as a prediction model, it's composed of fuzzy logic rules that provide a good lingustic representation.

Details

Package:	fugeR
Type:	Package
Version:	0.1
Date:	2012-07-11
License:	GPL (>= 2)
LazyLoad:	yes

This package allow you to use a genetic algorithm in order to find a fuzzy system that can be used as a prediction model.

`fugeR.run` find a fuzzy system. `fugeR.predict` make the prediction for the given input data.

Author(s)

Alexandre Bujard <alexandre.bujard@gmail.com>

References

http://library.epfl.ch/en/theses/?nr=2634

See Also

`fugeR.run`

Examples

```
## Not run:
#We use the iris dataset for this example
#We need to convert the output in a numeric format.
data(iris)
OUT <- data.matrix(iris[5])[,1]
fIris <- cbind(iris[1:4], OUT)
In <- fIris[1:4]
Out <- fIris[5]
```

```
#Launch the evolution, fugeR.run will return
#the best fuzzy system found during the evolution
fuzzySystem  <- fugeR.run( In,
                   Out,
                  generation=100, # Increase the number of generation for a better accuracy
                   population=100,
                   elitism=20,
                   verbose=TRUE,
                   threshold=NA,
                   sensiW=0.0,
                   speciW=0.0,
                   accuW=0.0,
                   rmseW=1.0,
                   maxRules=5,
                   maxVarPerRule=2,
                   labelsMf=3
)

#Plot the predicton given by the best fuzzy system found during the evolution
prediction <- fugeR.predict(fuzzySystem, In)
plot(prediction[[1]], ylim=c(1,max(unlist(Out))), col='blue', pch=21, axes=FALSE, ann=FALSE)
points(Out[[1]], col="red", pch=21)
axis(1)
axis(2, at=1:3, lab=c('setosa', 'versicolor', 'virginica'))
title(main='Fuzzy system prediction on Iris problem')
title(xlab="Cases")
title(ylab="Specie")
box()
legend(0.0, 3.0, c("Predicted","Actual"), cex=0.8,
       col=c("blue","red"), pch=c(21,21))

#Display the fuzzy system
fugeR.summary(fuzzySystem)

## End(Not run)
```

`fugeR.load` *Load a fuzzy system.*

Description

Load a fuzzy system.

Usage

```
fugeR.load(file="")
```

Arguments

`file`	[\"\"] A character string naming a file.

Details

Load a fuzzy system saved into a file with `fugeR.save`

Author(s)

Alexandre Bujard, HEIG-VD, Jul'2012

See Also

`fugeR.save`

Examples

```
##
##
## Not run:
fis <- fugeR.run (
In,
Out,
generation=100,
population=200,
elitism=40,
verbose=TRUE,
threshold=0.5,
sensiW=1.0,
speciW=1.0,
accuW=0.0,
rmseW=1.0,
maxRules=10,
maxVarPerRule=2,
labelsMf=2
)

fugeR.save( fis, file=\'./myFis.R\' )

savedFis <- fugeR.load( file=\'./myFis.R\' )

## End(Not run)
```

---

fugeR.predict *Compute the prediction of a fuzzy system for the given input data.*

Description

Compute the prediction of a fuzzy system for the given input data.

Usage

```
fugeR.predict(fuzzySystem, dataset)
```

Arguments

fuzzySystem	[NULL] The fuzzy system to use for computing the prediction.
dataset	[NULL] The data to use.

Value

prediction, A data.frame containing the predictions.

Note

The dataset must contain the same headers (in the same order) that the data used to find the system with fugeR.run.

Author(s)

Alexandre Bujard, HEIG-VD, Jul'2012

See Also

fugeR.run

Examples

```
##
##
## Not run:
#We use the iris dataset for this example
#We need to convert the output in a numeric format.
data(iris)
OUT <- data.matrix(iris[5])[,1]
fIris <- cbind(iris[1:4], OUT)
In <- fIris[1:4]
Out <- fIris[5]

#Launch the evolution, fugeR.run will return
#the best fuzzy system found during the evolution
fuzzySystem  <- fugeR.run( In,
                    Out,
                  generation=100, # Increase the number of generation for a better accuracy
                    population=100,
                    elitism=20,
                    verbose=TRUE,
                    threshold=NA,
                    sensiW=0.0,
                    speciW=0.0,
                    accuW=0.0,
                    rmseW=1.0,
                    maxRules=5,
                    maxVarPerRule=2,
                    labelsMf=3
)
```

```
#Plot the predicton given by the best fuzzy system found during the evolution
prediction <- fugeR.predict(fuzzySystem, In)
plot(prediction[[1]], ylim=c(1,max(unlist(Out))), col='blue', pch=21, axes=FALSE, ann=FALSE)
points(Out[[1]], col="red", pch=21)
axis(1)
axis(2, at=1:3, lab=c('setosa', 'versicolor', 'virginica'))
title(main='Fuzzy system prediction on Iris problem')
title(xlab="Cases")
title(ylab="Specie")
box()
legend(0.0, 3.0, c("Predicted","Actual"), cex=0.8,
       col=c("blue","red"), pch=c(21,21))

#Display the fuzzy system
fugeR.summary(fuzzySystem)

## End(Not run)
```

`fugeR.run` *R based Fuzzy logic evolutionary algorithm...*

Description

R based evolutionary algorithm for finding fuzzy systems

Usage

```
fugeR.run(data, labels, maxRules=4,
    maxVarPerRule=3, labelsMf=2, population=200,
    elitism=NA, mutation=0.01, generation=100,
    sensiW=1, speciW=1, accuW=0, threshold=0.5,
    rmseW=0.2, verbose=FALSE)
```

Arguments

`data`	[NULL] Data frame to be used for training (only numeric values are supported).
`labels`	[NULL] Labels of `data` (only numeric values are supported).
`maxRules`	[4] Maximum number of rule.
`maxVarPerRule`	[3] Maximum number of input variable per rule.
`labelsMf`	[2] Number of singleton for output variable membership function.
`population`	[200] The population size.
`elitism`	[NA] The number of chromosomes that are kept into the next generation. By default is about 20% of the population size.
`mutation`	[0.01] The chance that a gene in the chromosome mutates.
`generation`	[100] The number of generation made by the genetic algorithm.

sensiW	[1.0] The weight of the sensitivity in the fitness function.
speciW	[1.0] The weight of the specificity in the fitness function.
accuW	[0.0] The weight of the accuracy in the fitness function.
threshold	[0.5] The threshold to apply in order to calculate sensitivity, specificity and accuracy.
rmseW	[0.2] The weight of the "root mean square error" between labels and values predicted by the fuzzy system.
verbose	[FALSE] If true the algorithm will be more verbose. By default False.

Details

A machine learning algorithm for fuzzy system.
This function use a genetic algorithm in order to construct a fuzzy system able to fit the values given as `labels`. The `data` and `labels` are used has learning data.

This is a fuzzy system evolutionnary algorithm. A genetic algorithm is used to find a fuzzy system able to fit the the data given as labels.
The genetic algorithm generate a random `population` of fuzzy system. At each `generation` all the fuzzy system are tested with the data. Their predictions are then compared with the labels and a "performance" is given at each system. The top best system (`elitsm` are taken without modification for the next generation. The population is then used to generate the population for the next generation using crossover and mutation. At the end of the process (at the last generation) the fuzzy system that obtained the best performance is returned.

Value

`fis`, A list containing the logs of the evolution, the peformances of the best system and its description.

inputVarIds	The IDs of the variable used in the fuzzy system
inputMfIds	The IDs of the membership function used by each variable in the fuzzy system
inputMfs	The values used to caclculate the membership functions
outputVarIds	The IDs of each output variable of each rule
outputMfIds	The IDs of the membership function used by each output variable
outputMfs	the value used to compute the membership functions of the output variables
fitness	The fitness value reached by the best fuzzy system
mse	The Mean Square Error of the best fuzzy system
rmse	The Root Mean Square Error between labels and the prediction made by the best fuzzy system
accu	The accuracy of the prediction made by the best fuzzy system (only if a threshold different of NA was given as argument)
sensi	The sensitivity of the prediction made by the best fuzzy system (only if a threshold different of NA was given as argument)
speci	The specificity of the prediction made by the best fuzzy system (only if a threshold different of NA was given as argument)
evo	A list containing the evolution logs

Author(s)

Alexandre Bujard, HEIG-VD, Jul'2012

See Also

fugeR.sfRun fugeR.predictfugeR.summary fugeR.save fugeR.load

Examples

```
##
##
## Not run:
#We use the iris dataset for this example
#We need to convert the output in a numeric format.
data(iris)
OUT <- data.matrix(iris[5])[,1]
fIris <- cbind(iris[1:4], OUT)
In <- fIris[1:4]
Out <- fIris[5]

#Launch the evolution, fugeR.run will return
#the best fuzzy system found during the evolution
fuzzySystem  <- fugeR.run( In,
                   Out,
                  generation=100, # Increase the number of generation for a better accuracy
                   population=100,
                   elitism=20,
                   verbose=TRUE,
                   threshold=NA,
                   sensiW=0.0,
                   speciW=0.0,
                   accuW=0.0,
                   rmseW=1.0,
                   maxRules=5,
                   maxVarPerRule=2,
                   labelsMf=3
)

#Plot the predicton given by the best fuzzy system found during the evolution
prediction <- fugeR.predict(fuzzySystem, In)
plot(prediction[[1]], ylim=c(1,max(unlist(Out))), col='blue', pch=21, axes=FALSE, ann=FALSE)
points(Out[[1]], col="red", pch=21)
axis(1)
axis(2, at=1:3, lab=c('setosa', 'versicolor', 'virginica'))
title(main='Fuzzy system prediction on Iris problem')
title(xlab="Cases")
title(ylab="Specie")
box()
legend(0.0, 3.0, c("Predicted","Actual"), cex=0.8,
       col=c("blue","red"), pch=c(21,21))

#Display the fuzzy system
```

```
fugeR.summary(fuzzySystem)

## End(Not run)
```

---

fugeR.save *Save a fuzzy system into a file.*

Description

Save a fuzzy system into a file.

Usage

```
fugeR.save(fuzzySystem, file="")
```

Arguments

fuzzySystem	[NULL] The fuzzy system to save.
file	[\"\"] A character string naming a file.

Author(s)

Alexandre Bujard, HEIG-VD, Jul'2012

See Also

fugeR.load

Examples

```
##
##
## Not run:
fis <- fugeR.run (
In,
Out,
generation=100,
population=200,
elitism=40,
verbose=TRUE,
threshold=0.5,
sensiW=1.0,
speciW=1.0,
accuW=0.0,
rmseW=1.0,
maxRules=10,
maxVarPerRule=2,
labelsMf=2
)
```

```
fugeR.save( fis, file=\'./myFis.R\' )

savedFis <- fugeR.load( file=\'./myFis.R\' )

## End(Not run)
```

---

fugeR.sfRun *The parallel version of fugeR...*

Description

The parallel version of fugeR.run using snowfall.

Usage

```
fugeR.sfRun(data, labels, maxRules=4,
    maxVarPerRule=3, labelsMf=2, population=200,
    elitism=NA, mutation=0.01, generation=100,
    sensiW=1, speciW=1, accuW=0, threshold=0.5,
    rmseW=0.2, verbose=FALSE, path, rep=300,
    parallel=FALSE, cpus=1)
```

Arguments

data	[NULL] Data frame to be used for training (only numeric values are supported).
labels	[NULL] Labels of data (only numeric values are supported).
maxRules	[4] Maximum number of rule.
maxVarPerRule	[3] Maximum number of input variable per rule.
labelsMf	[2] Number of singleton for output variable membership function.
population	[200] The population size.
elitism	[NA] The number of chromosomes that are kept into the next generation. By default is about 20% of the population size.
mutation	[0.01] The chance that a gene in the chromosome mutates.
generation	[100] The number of generation made by the genetic algorithm.
sensiW	[1.0] The weight of the sensitivity in the fitness function.
speciW	[1.0] The weight of the specificity in the fitness function.
accuW	[0.0] The weight of the accuracy in the fitness function.
threshold	[0.5] The threshold to apply in order to calculate sensitivity, specificity and accuracy.
rmseW	[0.2] The weight of the "root mean square error" between labels and values predicted by the fuzzy system.
verbose	[FALSE] If true the algorithm will be more verbose. By default False.

path	[NULL] THe path where to save the fuzzy systems.
rep	[300] Number fuzzy system to find.
parallel	[TRUE] Logical value indicating if the function can run in parralel
cpus	[1] number of cpus that can be used

Details

The parallel version of fugeR.run. Will launch fugeR.run a number of times given as argument. This function use snowfall package in order to take benefit of mutli-core computers.

fugeR.sfRun sould be used when you want to repeat an experience many times.
This is usefull when you are searching the good parameters (maxRules, macVarPerRule) for a problem. fugeR.sfRun will launch fugeR.run and test the obtained system. It automatically resamples the data using bootstrapping method.
For example if the argument rep has the value 1000 and the number of sample in data is 100. FugeR.sfRun resample the data with replacement with the size of the resample equal to 100 (the size of the original data set) this constitute the training set, the samples that were not picked are taken to create the validation set. FugeR.run is then called with the training set and the obtained fuzzy systems is tested on the validation set. If rep value was 1000, this operation is repeated 1000 times.
FugeR.sfRun saves every systems in the directory specified by path and return a resume of the performance obtained by each system on their training and validation set.

Value

res, A data.frame of size rep containing the performance of each fuzzy system on training and validation set.

Author(s)

Alexandre Bujard, HEIG-VD, Jul'2012

See Also

fugeR.run fugeR.predictfugeR.summary fugeR.save fugeR.load

Examples

```
##
##
## Not run:
expResume <- fugeR.sfRun (
In,
Out,
generation=100,
population=200,
elitism=40,
verbose=TRUE,
threshold=0.5,
sensiW=1.0,
```

```
speciW=1.0,
accuW=0.0,
rmseW=1.0,
maxRules=10,
maxVarPerRule=2,
labelsMf=2,
path=\'./exp\',
rep=100,
parallel=TRUE,
cpus=2
)

## End(Not run)
```

`fugeR.summary` *Summarize a fuzzy system.*

Description

Summarize a fuzzy system.

Usage

```
fugeR.summary(fuzzySystem)
```

Arguments

`fuzzySystem`	[NULL] The fuzzy system to show.

Details

Show the text description of a fuzzy system in a human readable form.

Author(s)

Alexandre Bujard, HEIG-VD, Jul'2012

See Also

`fugeR.run`

Examples

```
##
##
## Not run:
fis <- fugeR.run (
In,
Out,
generation=100,
```

```
population=200,
elitism=40,
verbose=TRUE,
threshold=0.5,
sensiW=1.0,
speciW=1.0,
accuW=0.0,
rmseW=1.0,
maxRules=10,
maxVarPerRule=2,
labelsMf=2
)

fugeR.summary(fis)

## End(Not run)
```

Index