Package ‘EM.Fuzzy’

October 12, 2022

Type Package

Title EM Algorithm for Maximum Likelihood Estimation by Non-Precise Information

Version 1.0

Date 2018-08-08

Author Abbas Parchami (Department of Statistics, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman, Iran)

Maintainer Abbas Parchami <parchami@uk.ac.ir>

Description The EM algorithm is a powerful tool for computing maximum likelihood estimates with incomplete data. This package will help to applying EM algorithm based on triangular and trapezoidal fuzzy numbers (as two kinds of incomplete data). A method is proposed for estimating the unknown parameter in a parametric statistical model when the observations are triangular or trapezoidal fuzzy numbers. This method is based on maximizing the observed-data likelihood defined as the conditional probability of the fuzzy data; for more details and formulas see Denoeux (2011) <doi:10.1016/j.fss.2011.05.022>.

License LGPL (>= 3)

Imports FuzzyNumbers, DISTRIB

NeedsCompilation no

Repository CRAN

Date/Publication 2018-08-16 09:30:08 UTC

R topics documented:

EM.Fuzzy-package — *EM Algorithm for Maximum Likelihood Estimation by Non-Precise Information*

Description

The main goal of this package is easy estimation of the unknown parameter of a continues distribution by EM algorithm where the observed data are fuzzy rather than crisp. This package contains two major functions: (1) the function `EM.Triangular` works by Triangular Fuzzy Numbers (TFNs), and (2) the function `EM.Trapezoidal` works by Trapezoidal Fuzzy Numbers (TrFNs).

Author(s)

Abbas Parchami

References

Denoeux, T. (2011) Maximum likelihood estimation from fuzzy data using the EM algorithm, Fuzzy Sets and Systems 183, 72-91.

Gagolewski, M., Caha, J. (2015) FuzzyNumbers Package: Tools to deal with fuzzy numbers in R. R package version 0.4-1, https://cran.r-project.org/web/packages=FuzzyNumbers

Gagolewski, M., Caha, J. (2015) A guide to the FuzzyNumbers package for R (FuzzyNumbers version 0.4-1) http://FuzzyNumbers.rexamine.com

Examples

```
library(FuzzyNumbers)
library(DISTRIB, warn.conflicts = FALSE)

# Let us we are going to estimation the unknown mean of Normal population with known variance
# (e.g, sd(X) = 0.5) on the basis of 11 trapezoidal fuzzy numbers (which we simulate them in
# bellow for simplification).
n = 11
set.seed(1000)
c1 = rnorm(n, 10,.5)
c2 = rnorm(n, 10,.5)
for(i in 1:n) {if (c1[i] > c2[i]) { zarf <- c1[i]; c1[i] <- c2[i]; c2[i] <- zarf }}
round(c1,3); round(c2,3)
c1 <= c2
l = runif(n, 0,1); round(l,3)
u = runif(n, 0,1); round(u,3)


EM.Trapezoidal(T.dist="norm", T.dist.par=c(NA,0.5), par.space=c(-5,30), c1, c2, l, u,
    start=4, ebs=.0001, fig=2)
```

EM.Trapezoidal *MLE by EM algorithm based on Trapezoidal Fuzzy Data*

Description

This function can easily obtain Maximum Likelihood Estimation (MLE) for the unknown one-dimensional parameter on the basis of Trapezoidal Fuzzy observation.

Usage

```
EM.Trapezoidal(T.dist, T.dist.par, par.space, c1, c2, l, u, start, ebs=0.001,
 fig = 2)
```

Arguments

`T.dist`	the distribution name of the random variable is determined by characteristic element `T.dist`. The names of distributions is similar to `stats` package.
`T.dist.par`	a vector of distribution parameters with considered ordering in `stats` package. If `T.dist` has only one parameter (which obviously is unknown) the user must be considered `T.dist.par=NA`. Also, it may be `T.dist` has two parameters which one of them is unknown and another known. In such cases, the user must be considered `T.dist.par = c(NA, known parameter` where the first parameter is unknown, and `T.dist.par = c(known parameter, NA` where the second parameter is unknown. See bellow examples.
`par.space`	an interval which is a subset / subinterval of the parameter space and it must be contain the true value of unknown parameter.
`c1`	a vector with `length(c) = n` from the first point of the core-values of TrFNs.
`c2`	a vector with `length(c) = n` from the last point of the core-values of TrFNs. Therefore, it is obvious that $c1 <= c2$.
`l`	a vector with `length(c) = n` from the left spreads of TrFNs.
`u`	a vector with `length(c) = n` from the right spreads of TrFNs.
`start`	a real number from `par.space` which EM algorithm must be started / worked with this start point.
`ebs`	a real positive small number (e.g., 0.01, 0.001 or 0.1^6) which determine the accuracy of EM algorithm in estimation of unknown parameter.
`fig`	a numeric argument which can tack only values 0, 1 or 2. If `fig = 0`, the result of EM algorithm will not contains any figure. If `fig = 1`, then the membership functions of TrFNs will be shown in a figure with different colors. If `fig = 2`, then the membership functions of TrFNs will be shown in a figure with the curve of estimated probability density function (p.d.f.) on the basis of maximum likelihood estimation.

Value

The parameter computed / estimated in each iteration separately and also the computation of the following values can be asked directly.

- `MLE` — the value of maximum likelihood estimated for unknown parameter by EM algorithm based on TrFNs.
- `parameter.vector` — a vector of the ML estimated parameter for unknown parameter in algorithm which its first elements `start` and the last element is `MLE`.
- `Iter.Num` — the number of EM algorithm iterations.

Note

In using this package it must be noted that:

(1) The sample size of TrFNs must be less than 16. This package is able to work with small sample sizes ($n \leq 15$) and can be extended by the user if needs.

(2) Considering a suitable interval for `par.space` is very important to obtain a true result for EM algorithm. It must be mentioned that this interval must be a sub-interval of the parameter space and the user must check the result of algorithm (MLE). It means that if the obtained MLE (by `EM.Trapezoidal`) overlay on the boundary of `par.space`, then the result is not acceptable and the EM algorithm must be repeated once again with a wider `par.space`.

(3) This package is able to work for continuous distributions with one or two parameter which only one of them is unknown and the user wants to estimate it based on TrFNs.

See Also

DISTRIB FuzzyNumbers

Examples

```
library(FuzzyNumbers)
library(DISTRIB, warn.conflicts = FALSE)

# Example 1: Estimation the unknown mean of Normal population with known variance (e.g,
# var=0.5^2) based of Trapezoidal FNs.
n = 2
set.seed(1000)
c1 = rnorm(n, 10,.5)
c2 = rnorm(n, 10,.5)
for(i in 1:n) {if (c1[i] > c2[i]) { zarf <- c1[i]; c1[i] <- c2[i]; c2[i] <- zarf }}
round(c1,3); round(c2,3)
c1 <= c2
l = runif(n, 0,1); round(l,3)
u = runif(n, 0,1); round(u,3)

EM.Trapezoidal(T.dist="norm", T.dist.par=c(NA,0.5), par.space=c(-5,30), c1, c2, l, u, start=4,
ebs=.1, fig=2)
```

```
# Example 2:
n = 4
set.seed(10)
c1 = rexp(n, 2)
c2 = rexp(n, 2)
for(i in 1:n) {if (c1[i] > c2[i]) { zarf <- c1[i]; c1[i] <- c2[i]; c2[i] <- zarf }}
round(c1,3); round(c2,3)
c1 <= c2
l = runif(n, 0,1); round(l,3)
u = runif(n, 0,2); round(u,3)

EM.Trapezoidal(T.dist="exp", T.dist.par=NA, par.space=c(.1,20), c1, c2, l, u, start=7,
    ebs=.001)


# Example 3: Estimation the unknown standard deviation of Normal population with known
#  mean (e.g, mean=7) based of Trapezoidal FNs.
n = 10
set.seed(123)
c1 = rnorm(n, 4,1)
c2 = rnorm(n, 4,1)
for(i in 1:n) {if (c1[i] > c2[i]) { zarf <- c1[i]; c1[i] <- c2[i]; c2[i] <- zarf }}
round(c1,3); round(c2,3)
c1 <= c2
l = runif(n, 0,.5); round(l,3)
u = runif(n, 0,.75); round(u,3)

EM.Trapezoidal(T.dist="norm", T.dist.par=c(4,NA), par.space=c(0,40), c1, c2, l, u, start=1,
    ebs=.0001, fig=2)


# Example 4: Estimation alpha parameter in Beta distribution.
n = 4
set.seed(12)
c1 = rbeta(n, 2,1)
c2 = rbeta(n, 2,1)
for(i in 1:n) {if (c1[i] > c2[i]) { zarf <- c1[i]; c1[i] <- c2[i]; c2[i] <- zarf }}
round(c1,3); round(c2,3)
c1 <= c2
l = rbeta(n, 1,1); round(l,3)
u = rbeta(n, 1,1); round(u,3)

EM.Trapezoidal(T.dist="beta", T.dist.par=c(NA,1), par.space=c(0,10), c1, c2, l, u, start=1,
    ebs=.01, fig=2)
```

EM.Triangular *MLE by EM algorithm based on Triangular Fuzzy Data*

Description

This function can easily obtain Maximum Likelihood Estimation (MLE) for the unknown one-dimensional parameter on the basis of Triangular Fuzzy observation.

Usage

```
EM.Triangular(T.dist, T.dist.par, par.space, c, l, u, start, ebs = 0.001,
 fig = 2)
```

Arguments

`T.dist`	the distribution name of the random variable is determined by characteristic element `T.dist`. The names of distributions is similar to `stats` package.
`T.dist.par`	a vector of distribution parameters with considered ordering in `stats` package. If `T.dist` has only one parameter (which obviously is unknown) the user must be considered `T.dist.par=NA`. Also, it may be `T.dist` has two parameters which one of them is unknown and another known. In such cases, the user must be considered `T.dist.par = c(NA, known parameter` where the first parameter is unknown, and `T.dist.par = c(known parameter, NA` where the second parameter is unknown. See bellow examples.
`par.space`	an interval which is a subset / subinterval of the parameter space and it must be contain the true value of unknown parameter.
`c`	a vector with `length(c) = n` from the core-values of TFNs.
`l`	a vector with `length(c) = n` from the left spreads of TFNs.
`u`	a vector with `length(c) = n` from the right spreads of TFNs.
`start`	a real number from `par.space` which EM algorithm must be started / worked with this start point.
`ebs`	a real positive small number (e.g., 0.01, 0.001 or 0.1^6) which determine the accuracy of EM algorithm in estimation of unknown parameter.
`fig`	a numeric argument which can tack only values 0, 1 or 2. If `fig = 0`, the result of EM algorithm will not contains any figure. If `fig = 1`, then the membership functions of TFNs will be shown in a figure. If `fig = 2`, then the membership functions of TFNs will be shown in a figure with the curve of estimated probability density function (p.d.f.) on the basis of maximum likelihood estimation.

Value

The parameter computed / estimated in each iteration separately and also the computation of the following values can be asked directly.

`MLE`	the value of maximum likelihood estimated for unknown parameter by EM algorithm based on TFNs.
`parameter.vector`	a vector of the ML estimated parameter for unknown parameter in algorithm which its first elements `start` and the last element is `MLE`.
`Iter.Num`	the number of EM algorithm iterations.

Note

In using this package it must be noted that:

(1) The sample size of TFNs must be less than 16. This package is able to work with small sample sizes ($n \leq 15$) and can be extended by the user if needs.

(2) Considering a suitable interval for `par.space` is very important to obtain a true result for EM algorithm. It must be mentioned that this interval must be a sub-interval of the parameter space and the user must check the result of algorithm (MLE). It means that if the obtained MLE (by `EM.Triangular`) overlay on the boundary of `par.space`, then the result is not acceptable and the EM algorithm must be repeated once again with a wider `par.space`.

(3) This package is able to work for continuous distributions with one or two parameter which only one of them is unknown and the user wants to estimate it based on TFNs.

See Also

DISTRIB FuzzyNumbers

Examples

```
library(FuzzyNumbers)
library(DISTRIB, warn.conflicts = FALSE)

# Example 1:
n = 2
set.seed(131)
c = rexp(n, 2);    round(c,3)
l = runif(n, 0,1); round(l,3)
u = runif(n, 0,2); round(u,3)

EM.Triangular(T.dist="exp", T.dist.par=NA, par.space=c(0,30), c, l, u, start=5, ebs=.1,
  fig=0)


  EM.Triangular(T.dist="exp", T.dist.par=NA, par.space=c(0,30), c, l, u, start=50, ebs=.001,
      fig=1) #Fast Convergence

  EM.Triangular(T.dist="exp", T.dist.par=NA, par.space=c(0,30), c, l, u, start=50, ebs=.1^6,
      fig=2)


    #Example 2:  Computing the mean and the standard deviation of 20 EM estimations:
    n = 15
    MLEs=c()
    for(j in 100:120){
                     print(j)
                     set.seed(j)
                     c = rexp(n, 2)
                     l = runif(n, 0,1)
                     u = runif(n, 0,2)
                     MLEs = c(MLEs, EM.Triangular(T.dist="exp", T.dist.par=NA,
                                                  par.space=c(0,30), c, l, u, start=5,
```

```
                                                    ebs=.01, fig=0)$MLE )
                    }

  MLEs       #  3.283703 2.475541 3.171026 ...
  mean(MLEs) # 2.263996
  sd(MLEs)   # 0.4952257
  hist(MLEs)


  # Example 3: Estimation the unknown mean of Normal population with known variance
  #  (e.g, var=1) based of TFNs.
  n = 5
  set.seed(100)
  c = rnorm(n, 10,1);    round(c,3)
  l = runif(n, 0,1); round(l,3)
  u = runif(n, 0,1); round(u,3)

EM.Triangular(T.dist="norm", T.dist.par=c(NA,1), par.space=c(-10,30), c, l, u, start=20,
    ebs=.001, fig=2)


# Example 4: Estimation the unknown standard deviation of Normal population with known
  #  mean (e.g, mean=7) based of TFNs.
  n = 10
  set.seed(123)
  c = rnorm(n, 7,2);    round(c,3)
  l = runif(n, 0,2.5); round(l,3)
  u = runif(n, 0,1); round(u,3)

EM.Triangular(T.dist="norm", T.dist.par=c(7,NA), par.space=c(0,10), c, l, u, start=5,
    ebs=.0001, fig=2)


  # Example 5: Estimation the unknown parameter b where  X ~ U(a=0,b).
  n = 15
  set.seed(101)
  c = runif(n, 0,5); round(c,3)
  l = runif(n, 0,1); round(l,3)
  u = runif(n, 0,1); round(u,3)

 b <- EM.Triangular(T.dist="unif", T.dist.par=c(0,NA), par.space=c(0,10), c, l, u,
                   start=5, ebs=.001, fig=2)$MLE
  print(b)
```

Index