

Package ‘trajeR’

October 14, 2022

Type Package

Title Group Based Modeling Trajectory

Description Find the probability and the trajectory of longitudinal mixture model. Methods used in the package refer to Nagin (2005), [doi:10.4159/9780674041318](https://doi.org/10.4159/9780674041318), Nagin, D. (2005). Group-Based Modeling of Development. Cambridge, MA: Harvard University Press. and Noel (2022), <https://orbilu.uni.lu/>, thesis.

Version 0.9.0.5

Date 2022-02-14

Author Cédric NOEL - Jang Schiltz

Maintainer Cédric NOEL <cedric.noel@univ-lorraine.fr>

License GPL (>= 2)

Imports Rcpp (>= 1.0.4.6), minpack.lm, numDeriv, ucminf, MASS, capushe

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.1.2

Encoding UTF-8

URL <https://github.com/gitedric/trajeR>

BugReports <https://github.com/gitedric/trajeR/issues>

Depends R (>= 2.10)

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-02-21 13:30:01 UTC

R topics documented:

adequacy	2
AvePP	3
ConfIntT	4
diffaitbeta	5
fait	5
GroupProb	6
GroupProfiles	7
OCC	7
plottrajeR	8
plottrajeR.Trajectory.BETA	9
plottrajeR.Trajectory.CNORM	10
plottrajeR.Trajectory.LOGIT	11
plottrajeR.Trajectory.NL	12
plottrajeR.Trajectory.POIS	13
plottrajeR.Trajectory.ZIP	14
print.Trajectory.BETA	15
print.Trajectory.CNORM	16
print.Trajectory.LOGIT	16
print.Trajectory.NL	17
print.Trajectory.POIS	17
print.Trajectory.ZIP	18
propAssign	19
trajeR	19
trajeR.BETA	22
trajeR.CNORM	24
trajeR.LOGIT	27
trajeR.NL	29
trajeR.POIS	31
trajeR.ZIP	33
trajeRAIC	35
trajeRBIC	36
trajeRSH	37
Index	38

adequacy	<i>Adequacy of the model</i>
----------	------------------------------

Description

Calculate the summary of the five methods : assignment proportion, average posterior probability, confidence interval, odds of Correct Classification.

Usage

```
adequacy(sol, Y, A, nb = 10000, alpha = 0.98)
```

Arguments

sol	Trajectory's object. An object of type Trajectory.
Y	Matrix. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.
nb	Integer. The numbers of repetitions in the bootstrap method.
alpha	Real. The degree of confidence of the interval.

Value

A table of reals. A table with 5 rows: the estimate probabilities, the two bounds of the confidence interval, the proportion of assignment, the Average Posterior Probability and the Odds of Correct Classification.

Examples

```
data = read.csv(system.file("extdata", "CNORM2gr.csv", package = "trajeR"))
data = as.matrix(data)
sol = trajeR(Y = data[, 2:6], A = data[, 7:11], degre = c(2,2), Model = "CNORM", Method = "EM")
adequacy(sol, Y = data[, 2:6], A = data[, 7:11])
```

AvePP

Average Posterior Probability

Description

Calculate the Average Posterior Probability. Average Posterior Probability (AvePP) is the average posterior probability of membership for each group for those individuals that were assigned to.

Usage

```
AvePP(sol, Y, A, X = NULL)
```

Arguments

sol	Trajectory's object. An object of type Trajectory.
Y	Matrix. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.
X	Matrix. An optional matrix that modifies the probability of belong to group. By default its value is a one column matrix with value 1.

Value

A vector of reals. The average posterior probability.

Examples

```
data = read.csv(system.file("extdata", "CNORM2gr.csv", package = "trajeR"))
data = as.matrix(data)
sol = trajeR(Y = data[, 2:6], A = data[, 7:11], degre = c(2,2), Model = "CNORM", Method = "EM")
AvePP(sol, Y = data[, 2:6], A = data[, 7:11])
```

 ConfIntT

Confidence interval

Description

Calculate the confidence interval of the probabilities with bootstrap method. We have to specify the number of the repetitions of bootstrap and the degree of confidence.

Usage

```
ConfIntT(sol, Y, A, nb = 10000, alpha = 0.98)
```

Arguments

sol	Trajectory's object. An object of type Trajectory.
Y	Matrix. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.
nb	An integer. The number of repetitions in the bootstrap method.
alpha	A number. The degree of confidence of the interval.

Value

A vector of reals. The two bounds of the confidence interval given a degree of confidence.

Examples

```
data = read.csv(system.file("extdata", "CNORM2gr.csv", package = "trajeR"))
data = as.matrix(data)
sol = trajeR(Y = data[, 2:6], A = data[, 7:11], degre = c(2,2), Model = "CNORM", Method = "EM")
ConfIntT(sol, Y = data[, 2:6], A = data[, 7:11])
```

diffaitbeta	<i>Differential</i>
-------------	---------------------

Description

Differential

Usage

```
diffaitbeta(betak, i, t, A, TCOV, fct, diffct)
```

Arguments

betak	Vector of intger.
i	Integer.
t	Real.
A	Matrix of real.
TCOV	Matrix of real.
fct	Function.
diffct	Function.

Value

real. Cumpute the value of the diferential function fct for individual i, time t and group k.

fait	<i>Function fait</i>
------	----------------------

Description

Function fait

Usage

```
fait(betak, i, t, A, TCOV, fct, diffct)
```

Arguments

betak	Vector of intger.
i	Integer.
t	Real.
A	Matrix of real.
TCOV	Matrix of real.
fct	Function.
diffct	Function.

Value

real. Compute the value of the function fct for individual i, time t and group k.

GroupProb	<i>Membership's probabilities</i>
-----------	-----------------------------------

Description

GroupProb calculate the membership probability of each value of the data.

Usage

```
GroupProb(Obj, Y, A, TCOV = NULL, X = NULL)
```

Arguments

Obj	Trajectory's object. A trajectory object that is return by trajeR function.
Y	Matrix. A real matrix. The data.
A	Matrix. A real matrix. The time variable.
TCOV	Matrix. A real matrix. Optional, by default the value is NULL. It contained the time dependent covariate.
X	Matrix. A real matrix. Optional, by default the value is NULL. It contained a covariate that modify the probability membership.

Value

a real matrix. For each individual i in the data, this matrix contained the membership probability of each group.

Examples

```
data = read.csv(system.file("extdata", "CNORM2gr.csv", package = "trajeR"))
data = as.matrix(data)
sol = trajeR(Y = data[, 2:6], A = data[, 7:11], degre = c(2,2), Model = "CNORM", Method = "EM")
GroupProb(sol, Y=data[, 2:6], A=data[, 7:11])
```

GroupProfiles	<i>Profiles of each group</i>
---------------	-------------------------------

Description

GroupProfiles calculate the profile of a group regarding covariate. It is a cross tabulation of individual level trajectory group assignments with individual level characteristic that might be associated with trajectory group membership.

Usage

```
GroupProfiles(sol, Y, A, X)
```

Arguments

sol	Trajectory's object. A object of type trajectory.
Y	Matirx. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.
X	Matrix. An optional matrix that modify the probability of belong to group. By default its value is a matrix with one column with value 1.

Value

A table of real.

Examples

```
data = read.csv(system.file("extdata", "CNORM2gr.csv", package = "trajeR"))
data = as.matrix(data)
sol = trajeR(Y = data[, 2:6], A = data[, 7:11], Risk = data[,12],
            degre = c(2,2), Model = "CNORM", Method = "L")
GroupProfiles(sol, Y = data[, 2:6], A = data[, 7:11], X = data[,12])
```

OCC	<i>Odds of Correct Classification</i>
-----	---------------------------------------

Description

Calculate Odds of Correct Classification. The Odds of Correct Classification for group k (OCC_j) is the ratio between the odds of a correct classification into group j on the basis of the posterior probability rule and the odds of correct assignment based on random assignments with the probability of assignment to group j is done with π_{ik} , the probability estimate by the model.

Usage

```
OCC(sol, Y, A)
```

Arguments

sol	Trajectory's object. An object of type Trajectory.
Y	Matrix. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.

Value

A vector of reals. The Odds of Correct Classification.

Examples

```
data = read.csv(system.file("extdata", "CNORM2gr.csv", package = "trajeR"))
data = as.matrix(data)
sol = trajeR(Y = data[, 2:6], A = data[, 7:11], degree = c(2,2), Model = "CNORM", Method = "EM")
OCC(sol, Y = data[, 2:6], A = data[, 7:11])
```

plotrajeR

plot trajectory

Description

plot trajectory

Usage

```
plotrajeR(Obj, ...)
```

Arguments

Obj	an object of class "Trajectory".
...	optional parameters

Value

a graphic.

Examples

```
data = read.csv(system.file("extdata", "CNORM2gr.csv", package = "trajeR"))
data = as.matrix(data)
sol = trajeR(Y = data[, 2:6], A = data[, 7:11], degree = c(2,2), Model = "CNORM", Method = "EM")
plotrajeR(sol)
```

```
plotrajeR.Trajectory.BETA
  plot BETA trajectory
```

Description

plot BETA trajectory

Usage

```
## S3 method for class 'Trajectory.BETA'
plotrajeR(
  Obj,
  plotcov = NULL,
  col = "black",
  Y = NULL,
  A = NULL,
  Risk = NULL,
  TCOV = NULL,
  mean = FALSE,
  alpha = 1,
  ...
)
```

Arguments

Obj	an object of class "Trajectory.LOGIT".
plotcov	an optional vector or matrix with the same length as the time period. Default value is NULL.
col	an optional vector. The vector of colors. It must contain a color for each trajectory and each points of groups. Its length is the double of the number of group. Default value is a grayscale.
Y	Matrix. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.
Risk	Matrix. An optional matrix that modify the probability of belong to group. By default its value is a matrix with one column with value 1.
TCOV	Matrix. An optional matrix containing the time covariate that influence the trajectory themselves. By default its value is NULL.
mean	an optional logical. Indicate if the mean of ech group and time value must be draw.
alpha	on optional real. Indicate the alpha channel of the points color.
...	optional parameters

Value

a graphic.

```
plotrajeR.Trajectory.CNORM
  plot CNORM trajectory
```

Description

plot CNORM trajectory

Usage

```
## S3 method for class 'Trajectory.CNORM'
plotrajeR(
  Obj,
  plotcov = NULL,
  col = "black",
  Y = NULL,
  A = NULL,
  Risk = NULL,
  mean = FALSE,
  alpha = 1,
  ...
)
```

Arguments

Obj	an object of class "Trajectory.CNORM".
plotcov	an optionnal vector or matrix with the same length as the time period. Default value is NULL.
col	an optionnal vector. The vecotr of colors. It must contain a color for each trajectory and each points of groups. Its length is the double of the number of group. Default valme is a grayscale.
Y	Matrix. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.
Risk	Matrix. An optional matrix that modify the probability of belong to group. By default its value is a matrix with one column with value 1.
mean	an optional logical. Indicate if the mean of ech group and time value must be draw.
alpha	on optionnal real. Indiciate the alpha channel of the points color.
...	optional parameters

Value

a graphic.

```
plotrajeR.Trajectory.LOGIT
  plot LOGIT trajectory
```

Description

plot LOGIT trajectory

Usage

```
## S3 method for class 'Trajectory.LOGIT'
plotrajeR(
  Obj,
  plotcov = NULL,
  dec = 1,
  col = "black",
  Y = NULL,
  A = NULL,
  Risk = NULL,
  mean = FALSE,
  alpha = 1,
  ...
)
```

Arguments

Obj	an object of class "Trajectory.LOGIT".
plotcov	an optional vector or matrix with the same length as the time period. Default value is NULL.
dec	an optional real. It precise the shift to draw the data points.
col	an optional vector. The vector of colors. It must contain a color for each trajectory and each points of groups. Its length is the double of the number of group. Default value is a grayscale.
Y	Matrix. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.
Risk	Matrix. An optional matrix that modify the probability of belong to group. By default its value is a matrix with one column with value 1.
mean	an optional logical. Indicate if the mean of ech group and time value must be draw.
alpha	on optional real. Indicate the alpha channel of the points color.
...	optional parameters

Value

a graphic.

plotrajeR.Trajectory.NL

plot Non Linear trajectory

Description

plot Non Linear trajectory

Usage

```
## S3 method for class 'Trajectory.NL'
plotrajeR(
  Obj,
  plotcov = NULL,
  col = "black",
  Y = NULL,
  A = NULL,
  Risk = NULL,
  mean = FALSE,
  alpha = 1,
  TCOV = NULL,
  ...
)
```

Arguments

Obj	an object of class "Trajectory.LOGIT".
plotcov	an optional vector or matrix with the same length as the time period. Default value is NULL.
col	an optional vector. The vector of colors. It must contain a color for each trajectory and each points of groups. Its length is the double of the number of group. Default value is a grayscale.
Y	Matrix. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.
Risk	Matrix. An optional matrix that modify the probability of belong to group. By default its value is a matrix with one column with value 1.
mean	an optional logical. Indicate if the mean of ech group and time value must be draw.
alpha	on optional real. Indicate the alpha channel of the points color.
TCOV	Matrix. An optional matrix containing the time covariate that influence the trajectory themselves. By default its value is NULL.
...	optional parameters

Value

a graphic.

```
plotrajeR.Trajectory.POIS
  plot POIS trajectory
```

Description

plot POIS trajectory

Usage

```
## S3 method for class 'Trajectory.POIS'
plotrajeR(
  Obj,
  plotcov = NULL,
  dec = 0,
  col = "black",
  Y = NULL,
  A = NULL,
  Risk = NULL,
  TCOV = NULL,
  mean = FALSE,
  alpha = 1,
  ...
)
```

Arguments

Obj	an object of class "Trajectory.POIS".
plotcov	an optional vector or matrix with the same length as the time period. Default value is NULL.
dec	an optional real. It precise the shift to draw the data points.
col	an optional vector. The vector of colors. It must contain a color for each trajectory and each points of groups. Its length is the double of the number of group. Default value is a grayscale.
Y	Matrix. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.
Risk	Matrix. An optional matrix that modify the probability of belong to group. By default its value is a matrix with one column with value 1.
TCOV	Matrix. An optional matrix containing the time covariate that influence the trajectory themselves. By default its value is NULL.
mean	an optional logical. Indicate if the mean of ech group and time value must be draw.
alpha	on optional real. Indicate the alpha channel of the points color.
...	optional parameters

Value

a graphic.

plotrajeR.Trajectory.ZIP
plot ZIP trajectory

Description

plot ZIP trajectory

Usage

```
## S3 method for class 'Trajectory.ZIP'
plotrajeR(
  Obj,
  plotcov = NULL,
  dec = 1,
  col = "black",
  Y = NULL,
  A = NULL,
  Risk = NULL,
  TCOV = NULL,
  mean = FALSE,
  alpha = 1,
  ...
)
```

Arguments

Obj	an object of class "Trajectory.LOGIT".
plotcov	an optional vector or matrix with the same length as the time period. Default value is NULL.
dec	an optional real. It precise the shift to draw the data points.
col	an optional vector. The vector of colors. It must contain a color for each trajectory and each points of groups. Its length is the double of the number of group. Default value is a grayscale.
Y	Matrix. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.
Risk	Matrix. An optional matrix that modify the probability of belong to group. By default its value is a matrix with one column with value 1.
TCOV	Matrix. An optional matrix containing the time covariate that influence the trajectory themselves. By default its value is NULL.

mean	an optional logical. Indicate if the mean of each group and time value must be drawn.
alpha	an optional real. Indicate the alpha channel of the points color.
...	optional parameters

Value

a graphic.

`print.Trajectory.BETA` *Print BETA*

Description

Print method for an object of class "Trajectory.BETA".

Usage

```
## S3 method for class 'Trajectory.BETA'  
print(x, ...)
```

Arguments

x	Trajectory's object. An object of class "Trajectory.BETA".
...	optional parameters

Value

The print of Obj.

Examples

```
data = read.csv(system.file("extdata", "BETA2gr.csv", package = "trajeR"))  
data = as.matrix(data)  
data[,2:6] = data[,2:6]*(nrow(data[,2:6])-1+0.5)/nrow(data[,2:6])  
sol = trajeR(Y = data[, 2:6], A = data[, 7:11], itermax = 50,  
            degre = c(2,2), degre.phi = c(1,1), Model = "BETA", Method = "L")  
sol
```

```
print.Trajectory.CNORM
    Print CNORM
```

Description

Print method for an object of class "Trajectory.CNORM".

Usage

```
## S3 method for class 'Trajectory.CNORM'
print(x, ...)
```

Arguments

x Trajectory's object. An object of class "Trajectory.CNORM".
 ... optional parameters

Value

The print of Obj.

Examples

```
data = read.csv(system.file("extdata", "CNORM2gr.csv", package = "trajeR"))
data = as.matrix(data)
sol = trajeR(Y = data[, 2:6], A = data[, 7:11], degree = c(2,2), Model = "CNORM", Method = "EM")
sol
```

```
print.Trajectory.LOGIT
    Print LOGIT
```

Description

Print mehtod for an object of class "Trajectory.LOGIT".

Usage

```
## S3 method for class 'Trajectory.LOGIT'
print(x, ...)
```

Arguments

x Trajectory's object. . An object of class "Trajectory.LOGIT".
 ... optional parameters

Value

The print of Obj.

Examples

```
data = read.csv(system.file("extdata", "LOGIT2gr.csv", package = "trajeR"))
data = as.matrix(data)
sol = trajeR(Y = data[, 2:6], A = data[, 7:11], degree = c(1,2), Model = "LOGIT", Method = "L")
sol
```

```
print.Trajectory.NL    print NL trajectory
```

Description

Print method for an object of class "Trajectory.NL".

Usage

```
## S3 method for class 'Trajectory.NL'
print(x, ...)
```

Arguments

x Trajectory's object. . An object of class "Trajectory.NL".
... optional parameters

Value

The print of Obj.

```
print.Trajectory.POIS Print POIS
```

Description

Print mehtod for an object of class "Trajectory.POIS".

Usage

```
## S3 method for class 'Trajectory.POIS'
print(x, ...)
```

Arguments

x Trajectory's object. . An object of class "Trajectory.POIS".
... optional parameters

Value

The print of Obj.

Examples

```
data = read.csv(system.file("extdata", "POIS2gr.csv", package = "trajeR"))
data = as.matrix(data)
sol = trajeR(Y = data[, 2:6], A = data[, 7:11],
             degre = c(2,2), Model = "POIS", Method = "L", hessian = FALSE)
sol
```

`print.Trajectory.ZIP` *Print ZIP*

Description

Print method for an object of class "Trajectory.ZIP".

Usage

```
## S3 method for class 'Trajectory.ZIP'
print(x, ...)
```

Arguments

`x` Trajectory's object. An object of class "Trajectory.ZIP".
`...` optional parameters

Value

The print of Obj.

Examples

```
data = read.csv(system.file("extdata", "ZIP2gr.csv", package = "trajeR"))
data = as.matrix(data)
sol = trajeR(Y = data[, 2:6], A = data[, 7:11],
             degre = c(1,2), degre.nu = c(1,1), Model = "ZIP", Method = "L")
sol
```

propAssign	<i>Assignment proportion</i>
------------	------------------------------

Description

Calculate the proportion of individuals in a given group. That is the ratio of the number of individuals in one group and all the individuals.

Usage

```
propAssign(sol, Y, A)
```

Arguments

sol	Trajectory's object. An object of type Trajectory.
Y	Matrix. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.

Value

A vector of real. The proportion.

Examples

```
data = read.csv(system.file("extdata", "CNORM2gr.csv", package = "trajeR"))
data = as.matrix(data)
sol = trajeR(Y = data[, 2:6], A = data[, 7:11], degree = c(2,2), Model = "CNORM", Method = "EM")
propAssign(sol, Y = data[, 2:6], A = data[, 7:11])
```

trajeR	<i>Fitting longitudinal mixture models</i>
--------	--

Description

trajeR is used to fit longitudinal mixture models. It used 3 types of mixture models : LOGIT, ZIP and censored Normal.

Usage

```
trajeR(
  Y,
  A,
  Risk = NULL,
  TCOV = NULL,
  degree = NULL,
```

```

degre.nu = 0,
degre.phi = 0,
Model,
Method = "L",
ssigma = FALSE,
ymax = max(Y, na.rm = TRUE) + 1,
ymin = min(Y, na.rm = TRUE) - 1,
hessian = TRUE,
itermax = 100,
paraminit = NULL,
ProbIRLS = TRUE,
refgr = 1,
fct = NULL,
diffct = NULL,
nbvar = NULL,
ng.nl = NULL,
nls.limiter = 50
)

```

Arguments

Y	Matrix. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.
Risk	Matrix. An optional matrix that modify the probability of belong to group. By default its value is a matrix with one column with value 1.
TCOV	Matrix. An optional matrix containing the time covariate that influence the trajectory themselves. By default its value is NULL.
degre	Vector of integer. The degree of every polynomial function.
degre.nu	Vector of integer. The degree of all Poisson part for a ZIP model.
degre.phi	Vector of integer. The degree of beta parametr for a BETA model.
Model	String. The model used. The value are LOGIT for a Logit Mixture model, CNORM for a Censored Normal Mixture Model or ZIP for Zero Inflated Poisson Mixture model.
Method	String. Determine the method used for find the parameters of the model. The value are L for the Maximum Likelihood Estimation, EM for Expectation Maximization method with quasi newton method inside, EMIWRLS for Expectation Maximization method with Iterative Weighted Least Square.
ssigma	Logical. By default its value is FALSE. For the CNORM model, indicate if we want the same sigma for all normal density function.
ymax	Real. For the CNORM model, indicate the maximum value of the data. It concern only the model with censored data. By default its value is the maximum value of the data plus 1.
ymin	Real. For the CNORM model, indicate the minimum value of the data. It concern only the model with censored data. By default its value is the maximum value of the data minus 1.

<code>hessian</code>	Logical. Indicate if we want calculate the hessian matrix. Default is FALSE. If the method use is Likelihood, the hessian is calculated by inverting the Information's Fisher Matrix. To avoid numerically singular matrix we find the pseudo inverse matrix by using the <code>ginv</code> function in the package MASS. If the method is EM or EMIWRLS, the hessian is calculated by using Louis method.
<code>itermax</code>	Integer. Indicate the maximal number of iteration for <code>optim</code> function or for the EM algorithm.
<code>paraminit</code>	Vector. The vector of initial parameters. By default trajeR calculate the initial value based of the range or the standard deviation.
<code>ProbIRLS</code>	Logical. Indicate the method to sue in the search of predictor's probability. If TRUE (by default) we use IRLS method and if FALSE we use optimization method.
<code>refgr</code>	Integer. The number of reference group. By default is 1.
<code>fct</code>	Function. The definition of the function <code>f</code> in the definition in nonlinear model.
<code>diffct</code>	Function. The differential of the function <code>f</code> in the nonlinear model.
<code>nbvar</code>	Integer. The number of variable in the nonlinear model.
<code>ng.nl</code>	Integer. The number of group for a non linear model.
<code>nls.limiter</code>	Integer. In the case of non linear model, the maximum number of iterations allowed.

Details

Models for trajeR is, by default, a polynomial regression of the time value parameters for each groups. The number fo group is controlled by the integer `ng`. We can spcecify the degree of the polynomial shape for each groups by the vector `degre`.

Value

return an object of class "Trajectory.LOGIT". The generic accessor functions `beta`, `delta`, `theta`, `sd`, `tab`, `Likelihood`, `ng`, `model` and `method` extract various useful features of the value returned by `trajeR`.

An object of class "Trajectory.LOGIT" is a list containing at least the following components:

`beta` a vector of the parameters `beta`.

`delta` a vector of the parameter `delta`. Only if we use time covariate.

`theta` a vector with the parameter `theta` if there exist a covariate `X` that modify the probability or the probability of group membership.

`sd` a vector of the standrad deviation of the parameters.

`tab` a matrix with all the parameters and standard deviation.

`Likelihood` a real with the Likelihhod obtnaied by the parameters.

`ng` a integer with the number of group.

`model` a string with the model used.

`method` a string with the method used.

Examples

```
## Not run:
load("data/dataNORM01.RData")
soll = trajeR(data[,1:5], data[,6:10], ng = 3, degre=c(2,2,2),
              Model="CNORM", Method = "L", ssigma = FALSE,
              hessian = TRUE)

## End(Not run)
```

trajeR.BETA

Internal function to fit Beta regression

Description

Internal function to fit Beta regression

Usage

```
trajeR.BETA(
  Y,
  A,
  X,
  TCOV,
  ng,
  nx,
  n,
  nbeta,
  nphi,
  nw,
  ntheta,
  period,
  degre,
  theta,
  beta,
  phi,
  delta,
  pi,
  Method,
  hessian,
  itermax,
  paraminit,
  EMIRLS,
  refgr
)
```

Arguments

Y	Matrix. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.
X	Matrix. An optional matrix that modify the probability of belong to group. By default its value is a matrix with one column with value 1.
TCOV	Matrix. An optional matrix containing the time covariate that influence the trajectory themselves. By default its value is NULL.
ng	Integer. The number of groups.
nx	Integer. The number of covariates.
n	Integer. Number of individuals.
nbeta	Vector of integers. Number of beta parameters for each group.
nphi	Vector of integers. Number of phi parameters for each group.
nw	Integer. Number of time dependent covariate.
ntheta	Vector of integers. Number of theta parameters for each group.
period	Integer.
degre	Vector of integer. The degree of every polynomial function.
theta	Vector of real. The parameter for calculated the group membership probability.
beta	Vector of real. The beta parameter.
phi	Vector of real. The phi parameter.
delta	Vector of real. The delta parameter.
pi	Vector of real. The group membership probability.
Method	String. Determine the method used for find the parameters of the model. The value are L for the Maximum Likelihood Estimation, EM for Expectation Maximization method with quasi newton method inside, EMIWRLS for Expectation Maximization method with Iterative Weighted Least Square.
hessian	Logical. Indicate if we want calculate the hessian matrix. Default is FALSE. If the method use is Likelihood, the hessian is calculated by inverting the Information's Fisher Matrix. To avoid numerically singular matrix we find the pseudo inverse matrix by using the <code>ginv</code> function in the package MASS. If the method is EM or EMIWRLS, the hessian is calculated by using Louis method.
itermax	Integer. Indicate the maximal number of iteration for <code>optim</code> function or for the EM algorithm.
paraminit	Vector. The vector of initial parameters. By default trajeR calculate the initial value based of the range or the standard deviation.
EMIRLS	Boolean. True if we use EMIRLS method.
refgr	Integer. The number of reference group. By default is 1.

Value

return a object of class Trajectory.NL

- beta - vector of the parameter beta.
- sigma - vector of the parameters sigma.
- delta - vector of the parameter delta. Only if we use time covariate.
- theta - vector with the parameter theta if there exist a coavriate X that modify the probability or the probability of group membership.
- sd - vector of the standard deviation of the parameters.
- tab - a matrix with all the parameters and standard deviation.
- Model - a string with the model used.
- groups - a integer with the number of group.
- Names - strings with the name of the parameters.
- Method - a string with the method used.
- Size - a integer with the number of individuals.
- Likelihood - a real with the Likelihood obtained by the parameters.
- Time - a vector with the first row of time values.
- degre - a vector with the degree of the polynomial shape.

trajeR.CNORM

Internal function to fit CNORM Model

Description

Internal function to fit CNORM Model

Usage

```
trajeR.CNORM(
  Y,
  A,
  X,
  TCOV,
  ng,
  nx,
  n,
  nbeta,
  nw,
  ntheta,
  period,
  degre,
  theta,
  beta,
```

```

    sigma,
    delta,
    pi,
    Method,
    sigma,
    ymax,
    ymin,
    hessian,
    itermax,
    paraminit,
    EMIRLS,
    refgr
)

```

Arguments

Y	Matrix. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.
X	Matrix. An optional matrix that modify the probability of belong to group. By default its value is a matrix with one column with value 1.
TCOV	Matrix. An optional matrix containing the time covariate that influence the trajectory themselves. By default its value is NULL.
ng	Integer. The number of groups.
nx	Integer. The number of covariates.
n	Integer. Number of individuals.
nbeta	Vector of integers. Number of beta parameters for each group.
nw	Integer. Number of time dependent covariate.
ntheta	Vector of integers. Number of theta parameters for each group.
period	Integer.
degre	Vector of integer. The degree of every polynomial function.
theta	Vector of real. The parameter for calculated the group membership probability.
beta	Vector of real. The beta parameter.
sigma	Vector of real. The sigma parameter.
delta	Vector of real. The delta parameter.
pi	Vector of real. The group membership probability.
Method	String. Determine the method used for find the parameters of the model. The value are L for the Maximum Likelihood Estimation, EM for Expectation Maximization method with quasi newton method inside, EMIWRLS for Expectation Maximization method with Iterative Weighted Least Square.
sigma	Logical. By default its value is FALSE. For the CNORM model, indicate if we want the same sigma for all normal density function.
ymax	Real. For the CNORM model, indicate the maximum value of the data. It concern only the model with censored data. By default its value is the maximum value of the data plus 1.

ymin	Real. For the CNORM model, indicate the minimum value of the data. It concern only the model with censored data. By default its value is the maximum value of the data minus 1.
hessian	Logical. Indicate if we want calculate the hessian matrix. Default is FALSE. If the method use is Likelihood, the hessian is calculated by inverting the Information's Fisher Matrix. To avoid numerically singular matrix we find the pseudo inverse matrix by using the <code>ginv</code> function in the package MASS. If the method is EM or EMIWRLS, the hessian is calculated by using Louis method.
itermax	Integer. Indicate the maximal number of iteration for <code>optim</code> function or for the EM algorithm.
paraminit	Vector. The vector of initial parameters. By default <code>trajeR</code> calculate the initial value based of the range or the standard deviation.
EMIRLS	Boolean. True if we use EMIRLS method.
refgr	Integer. The number of reference group. By default is 1.

Value

return a object of class `Trajectory.CNORM`

- beta - vector of the parameter beta.
- sigma - vector of the parameters sigma.
- delta - vector of the parameter delta. Only if we use time covariate.
- theta - vector with the parameter theta if there exist a covariate X that modify the probability or the probability of group membership.
- sd - vector of the standard deviation of the parameters.
- tab - a matrix with all the parameters and standard deviation.
- Model - a string with the model used.
- groups - a integer with the number of group.
- Names - strings with the name of the parameters.
- Method - a string with the method used.
- Size - a integer with the number of individuals.
- Likelihood - a real with the Likelihood obtained by the parameters.
- Time - a vector with the first row of time values.
- degre - a vector with the degree of the polynomial shape.
- min - a real with the minimum value for censored data.
- max - a real with the maximum value for censored data.

trajeR.LOGIT

*Internal function to fit LOGIT Model***Description**

Internal function to fit LOGIT Model

Usage

```
trajeR.LOGIT(
  Y,
  A,
  X,
  TCOV,
  ng,
  nx,
  n,
  nbeta,
  nw,
  ntheta,
  period,
  degre,
  theta,
  beta,
  delta,
  pi,
  Method,
  hessian,
  itermax,
  paraminit,
  EMIRLS,
  refgr
)
```

Arguments

Y	Matrix. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.
X	Matrix. An optional matrix that modify the probability of belong to group. By default its value is a matrix with one column with value 1.
TCOV	Matrix. An optional matrix containing the time covariate that influence the trajectory themselves. By default its value is NULL.
ng	Integer. The number of groups.
nx	Integer. The number of covariates.
n	Integer. Number of individuals.

<code>nbeta</code>	Vector of integers. Number of beta parameters for each group.
<code>nw</code>	Integer. Number of time dependent covariate.
<code>ntheta</code>	Vector of integers. Number of theta parameters for each group.
<code>period</code>	Integer.
<code>degre</code>	Vector of integer. The degree of every polynomial function.
<code>theta</code>	Vector of real. The parameter for calculated the group membership probability.
<code>beta</code>	Vector of real. The beta parameter.
<code>delta</code>	Vector of real. The delta parameter.
<code>pi</code>	Vector of real. The group membership probability.
<code>Method</code>	String. Determine the method used for find the parameters of the model. The value are L for the Maximum Likelihood Estimation, EM for Expectation Maximization method with quasi newton method inside, EMIWRLS for Expectation Maximization method with Iterative Weighted Least Square.
<code>hessian</code>	Logical. Indicate if we want calculate the hessian matrix. Default is FALSE. If the method use is Likelihood, the hessian is calculated by inverting the Information's Fisher Matrix. To avoid numerically singular matrix we find the pseudo inverse matrix by using the <code>ginv</code> function in the package MASS. If the method is EM or EMIWRLS, the hessian is calculated by using Louis method.
<code>itermax</code>	Integer. Indicate the maximal number of iteration for <code>optim</code> function or for the EM algorithm.
<code>paraminit</code>	Vector. The vector of initial parameters. By default trajeR calculate the initial value based of the range or the standard deviation.
<code>EMIRLS</code>	Boolean. True if we use EMIRLS method.
<code>refgr</code>	Integer. The number of reference group. By default is 1.

Value

return a object of class Trajectory.LOGIT

- `beta` - vector of the parameter beta.
- `delta` - vector of the parameter delta. Only if we use time covariate.
- `theta` - vector with the parameter theta if there exist a covariate X that modify the probability or the probability of group membership.
- `sd` - vector of the standard deviation of the parameters.
- `tab` - a matrix with all the parameters and standard deviation.
- `Model` - a string with the model used.
- `groups` - a integer with the number of group.
- `Names` - strings with the name of the parameters.
- `Method` - a string with the method used.
- `Size` - a integer with the number of individuals.
- `Likelihood` - a real with the Likelihood obtained by the parameters.
- `Time` - a vector with the first row of time values.
- `degre` - a vector with the degree of the polynomial shape.

trajeR.NL

*Internal function to fit Non Linear Model***Description**

Internal function to fit Non Linear Model

Usage

```
trajeR.NL(
  Y,
  A,
  X,
  TCOV,
  ng,
  nx,
  n,
  nbeta,
  nw,
  ntheta,
  period,
  degre,
  theta,
  beta,
  sigma,
  pi,
  Method,
  ssigma,
  hessian,
  itermax,
  paraminit,
  EMIRLS,
  refgr,
  fct,
  diffct,
  nls.limiter
)
```

Arguments

Y	Matrix. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.
X	Matrix. An optional matrix that modify the probability of belong to group. By default its value is a matrix with one column with value 1.
TCOV	Matrix. An optional matrix containing the time covariate that influence the trajectory themselves. By default its value is NULL.

ng	Integer. The number of groups.
nx	Integer. The number of covariates.
n	Integer. Number of individuals.
nbeta	Vector of integers. Number of beta parameters for each group.
nw	Integer. Number of time dependent covariate.
ntheta	Vector of integers. Number of theta parameters for each group.
period	Integer.
degre	Vector of integer. The degree of every polynomial function.
theta	Vector of real. The parameter for calculated the group membership probability.
beta	Vector of real. The beta parameter.
sigma	Vector of real. The sigma parameter.
pi	Vector of real. The group membership probability.
Method	String. Determine the method used for find the parameters of the model. The value are L for the Maximum Likelihood Estimation, EM for Expectation Maximization method with quasi newton method inside, EMIWRLS for Expectation Maximization method with Iterative Weighted Least Square.
ssigma	Logical. By default its value is FALSE. For the CNORM model, indicate if we want the same sigma for all normal density function.
hessian	Logical. Indicate if we want calculate the hessian matrix. Default is FALSE. If the method use is Likelihood, the hessian is calculated by inverting the Information's Fisher Matrix. To avoid numerically singular matrix we find the pseudo inverse matrix by using the ginv function in the package MASS. If the method is EM or EMIWRLS, the hessian is calculated by using Louis method.
itermax	Integer. Indicate the maximal number of iteration for optim function or for the EM algorithm.
paraminit	Vector. The vector of initial parameters. By default trajeR calculate the initial value based of the range or the standard deviation.
EMIRLS	Boolean. True if we use EMIRLS method.
refgr	Integer. The number of reference group. By default is 1.
fct	Function. The definition of the function f in the definition in nonlinear model.
diffct	Function. The differential of the function f in the nonlinear model.
nls.limiter	Integer. In the case of non linear model, the maximum number of iterations allowed.

Value

return a object of class Trajectory.NL

- beta - vector of the parameter beta.
- sigma - vector of the parameters sigma.
- delta - vector of the parameter delta. Only if we use time covariate.

- theta - vector with the parameter theta if there exist a covariate X that modify the probability or the probability of group membership.
- sd - vector of the standard deviation of the parameters.
- tab - a matrix with all the parameters and standard deviation.
- Model - a string with the model used.
- groups - a integer with the number of group.
- Names - strings with the name of the parameters.
- Method - a string with the method used.
- Size - a integer with the number of individuals.
- Likelihood - a real with the Likelihood obtained by the parameters.
- Time - a vector with the first row of time values.
- degre - a vector with the degree of the polynomial shape.
- fct - the defintion of the function used int this model.

trajeR.POIS

Internal function to fit poisson Model

Description

Internal function to fit poisson Model

Usage

```
trajeR.POIS(  
  Y,  
  A,  
  X,  
  TCOV,  
  ng,  
  nx,  
  n,  
  nbeta,  
  nw,  
  ntheta,  
  period,  
  degre,  
  theta,  
  beta,  
  delta,  
  pi,  
  Method,  
  hessian,  
  itermax,
```

```

    paraminit,
    EMIRLS,
    refgr
)

```

Arguments

Y	Matrix. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.
X	Matrix. An optional matrix that modify the probability of belong to group. By default its value is a matrix with one column with value 1.
TCOV	Matrix. An optional matrix containing the time covariate that influence the trajectory themselves. By default its value is NULL.
ng	Integer. The number of groups.
nx	Integer. The number of covariates.
n	Integer. Number of individuals.
nbeta	Vector of integers. Number of beta parameters for each group.
nw	Integer. Number of time dependent covariate.
ntheta	Vector of integers. Number of theta parameters for each group.
period	Integer.
degre	Vector of integer. The degree of every polynomial function.
theta	Vector of real. The parameter for calculated the group membership probability.
beta	Vector of real. The beta parameter.
delta	Vector of real. The delta parameter.
pi	Vector of real. The group membership probability.
Method	String. Determine the method used for find the parameters of the model. The value are L for the Maximum Likelihood Estimation, EM for Expectation Maximization method with quasi newton method inside, EMIWRLS for Expectation Maximization method with Iterative Weighted Least Square.
hessian	Logical. Indicate if we want calculate the hessian matrix. Default is FALSE. If the method use is Likelihood, the hessian is calculated by inverting the Information's Fisher Matrix. To avoid numerically singular matrix we find the pseudo inverse matrix by using the <code>ginv</code> function in the package MASS. If the method is EM or EMIWRLS, the hessian is calculated by using Louis method.
itermax	Integer. Indicate the maximal number of iteration for <code>optim</code> function or for the EM algorithm.
paraminit	Vector. The vector of initial parameters. By default trajeR calculate the initial value based of the range or the standard deviation.
EMIRLS	Boolean. True if we use EMIRLS method.
refgr	Integer. The number of reference group. By default is 1.

Value

return a object of class Trajectory.Pois

- beta - vector of the parameter beta.
- delta - vector of the parameter delta. Only if we use time covariate.
- theta - vector with the parameter theta if there exist a covariate X that modify the probability or the probability of group membership.
- sd - vector of the standard deviation of the parameters.
- tab - a matrix with all the parameters and standard deviation.
- Model - a string with the model used.
- groups - a integer with the number of group.
- Names - strings with the name of the parameters.
- Method - a string with the method used.
- Size - a integer with the number of individuals.
- Likelihood - a real with the Likelihood obtained by the parameters.
- Time - a vector with the first row of time values.
- degre - a vector with the degree of the polynomial shape for the Poisson part.

trajeR.ZIP

Internal function to fit ZIP Model

Description

Internal function to fit ZIP Model

Usage

```
trajeR.ZIP(  
  Y,  
  A,  
  X,  
  TCOV,  
  ng,  
  nx,  
  n,  
  nbeta,  
  nw,  
  ntheta,  
  period,  
  degre,  
  degre.nu,  
  theta,  
  beta,
```

```

    nu,
    delta,
    pi,
    Method,
    hessian,
    itermax,
    paraminit,
    EMIRLS,
    refgr
)

```

Arguments

Y	Matrix. A matrix containing the variables in the model.
A	Matrix. A matrix containing the time variable data.
X	Matrix. An optional matrix that modify the probability of belong to group. By default its value is a matrix with one column with value 1.
TCOV	Matrix. An optional matrix containing the time covariate that influence the trajectory themselves. By default its value is NULL.
ng	Integer. The number of groups.
nx	Integer. The number of covariates.
n	Integer. Number of individuals.
nbeta	Vector of integers. Number of beta parameters for each group.
nw	Integer. Number of time dependent covariate.
ntheta	Vector of integers. Number of theta parameters for each group.
period	Integer.
degre	Vector of integer. The degree of every polynomial function.
degre.nu	Vector of integer. The degree of all Poisson part for a ZIP model.
theta	Vector of real. The parameter for calculated the group membership probability.
beta	Vector of real. The beta parameter.
nu	Vector of real. The nu parameter.
delta	Vector of real. The delta parameter.
pi	Vector of real. The group membership probability.
Method	String. Determine the method used for find the parameters of the model. The value are L for the Maximum Likelihood Estimation, EM for Expectation Maximization method with quasi newton method inside, EMIWRLS for Expectation Maximization method with Iterative Weighted Least Square.
hessian	Logical. Indicate if we want calculate the hessian matrix. Default is FALSE. If the method use is Likelihood, the hessian is calculated by inverting the Information's Fisher Matrix. To avoid numerically singular matrix we find the pseudo inverse matrix by using the <code>ginv</code> function in the package MASS. If the method is EM or EMIWRLS, the hessian is calculated by using Louis method.

itermax	Integer. Indicate the maximal number of iteration for optim function or for the EM algorithm.
paraminit	Vector. The vector of initial parameters. By default trajeR calculate the initial value based of the range or the standard deviation.
EMIRLS	Boolean. True if we use EMIRLS method.
refgr	Integer. The number of reference group. By default is 1.

Value

return a object of class Trajectory.ZIP

- beta - vector of the parameter beta.
- delta - vector of the parameter delta. Only if we use time covariate.
- theta - vector with the parameter theta if there exist a coavriate X that modify the probability or the probability of group membership.
- nu - vector of the parameters nu.
- sd - vector of the standard deviation of the parameters.
- tab - a matrix with all the parameters and standard deviation.
- Model - a string with the model used.
- groups - a integer with the number of group.
- Names - strings with the name of the parameters.
- Method - a string with the method used.
- Size - a integer with the number of individuals.
- Likelihood - a real with the Likelihood obtained by the parameters.
- Time - a vector with the first row of time values.
- degre - a vector with the degree of the polynomial shape for the Poisson part.
- degre.nu - a vector with the degree of the polynomial shape for the exceeded zero state.

trajeRAIC

AIC function to an trajectory object

Description

Calculate the AIC value to an trajectory object.

Usage

```
trajeRAIC(sol)
```

Arguments

sol Trajectory's object. An object of type trajectory.

Value

A real.

Examples

```
data = read.csv(system.file("extdata", "CNORM2gr.csv", package = "trajeR"))
data = as.matrix(data)
sol = trajeR(Y = data[, 2:6], A = data[, 7:11], degre = c(2,2), Model = "CNORM", Method = "EM")
trajeRAIC(sol)
```

trajeRBIC

BIC function to an trajectory object

Description

Calculate the BIC value to an trajectory object.

Usage

```
trajeRBIC(sol)
```

Arguments

sol Trajectory's object. An object of type trajectory.

Value

A real.

Examples

```
data = read.csv(system.file("extdata", "CNORM2gr.csv", package = "trajeR"))
data = as.matrix(data)
sol = trajeR(Y = data[, 2:6], A = data[, 7:11], degre = c(2,2), Model = "CNORM", Method = "EM")
trajeRBIC(sol)
```

`trajeRSH`*SH function to an trajectory object*

Description

Calculate the Slope Heuristic value to a list of trajectory objects.

Usage

```
trajeRSH(l)
```

Arguments

1 List. A list of objects of type trajectory.

Value

A vector of real.

Examples

```
data = read.csv(system.file("extdata", "CNORM2gr.csv", package = "trajeR"))
data = as.matrix(data)
degre = list(c(2,2), c(1,1), c(1,2), c(2,1), c(0,0), c(0,1), c(1,0), c(0,0), c(0,2), c(2,0))
sol = list()
for (i in 1:10){
  sol[[i]] = trajeR(Y = data[, 2:6], A = data[, 7:11],
                  degre = degre[[i]], Model = "CNORM", Method = "EM")
}
trajeRSH(sol)
```

Index

adequacy, [2](#)
AvePP, [3](#)

ConfIntT, [4](#)

diffaitbeta, [5](#)

fait, [5](#)

GroupProb, [6](#)
GroupProfiles, [7](#)

OCC, [7](#)

plotrajeR, [8](#)
plotrajeR.Trajectory.BETA, [9](#)
plotrajeR.Trajectory.CNORM, [10](#)
plotrajeR.Trajectory.LOGIT, [11](#)
plotrajeR.Trajectory.NL, [12](#)
plotrajeR.Trajectory.POIS, [13](#)
plotrajeR.Trajectory.ZIP, [14](#)
print.Trajectory.BETA, [15](#)
print.Trajectory.CNORM, [16](#)
print.Trajectory.LOGIT, [16](#)
print.Trajectory.NL, [17](#)
print.Trajectory.POIS, [17](#)
print.Trajectory.ZIP, [18](#)
propAssign, [19](#)

trajeR, [19](#)
trajeR.BETA, [22](#)
trajeR.CNORM, [24](#)
trajeR.LOGIT, [27](#)
trajeR.NL, [29](#)
trajeR.POIS, [31](#)
trajeR.ZIP, [33](#)
trajeRAIC, [35](#)
trajeRBIC, [36](#)
trajeRSH, [37](#)