Package: trajeR (via r-universe)

October 13, 2026

Type Package

Title Group Based Modeling Trajectory

Description Estimation of group-based trajectory models, including finite mixture models for longitudinal data, supporting censored normal, zero-inflated Poisson, logit, and beta distributions, using expectation-maximization and quasi-Newton methods, with tools for model selection, diagnostics, and visualization of latent trajectory groups, <doi: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 1.0

Date 2026-07-22

License GPL (>= 2)

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

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.3.3

Encoding UTF-8

URL https://codeberg.org/cidric/trajeR

BugReports https://codeberg.org/cidric/trajeR/issues

Depends R (>= 3.5)

Suggests knitr, rmarkdown, spelling, kableExtra

Language en-US

LazyData true

VignetteBuilder knitr

SystemRequirements C++17

Repository https://gitedric.r-universe.dev

Date/Publication 2026-08-10 17:41:42 UTC

RemoteUrl https://codeberg.org/cidric/trajer

RemoteRef HEAD

RemoteSha 05f517cefae56626b4101ce02acb685420236f70

Contents

adequacy . . . 3
AvePP . . . 4
confidenceInt . . . 4
ConfIntT . . . 5
data_CNORM . . . 6
data_CNORM_Censored . . . 7
data_LOGIT . . . 7
data_ZIP . . . 8
dataNORM01 . . . 8
diffaitbeta . . . 9
fait . . . 10
get_group_probabilities . . . 11
GroupProb . . . 12
GroupProfiles . . . 13
OCC . . . 13
plot_trajectory_mult . . . 14
plotrajeR . . . 15
plotrajeR.Trajectory.BETA . . . 15
plotrajeR.Trajectory.CNORM . . . 16
plotrajeR.Trajectory.LOGIT . . . 17
plotrajeR.Trajectory.NL . . . 18
plotrajeR.Trajectory.POIS . . . 19
plotrajeR.Trajectory.ZIP . . . 20
print.Trajectory.BETA . . . 21
print.Trajectory.CNORM . . . 22
print.Trajectory.LOGIT . . . 23
print.trajectory.mult . . . 23
print.Trajectory.NL . . . 24
print.Trajectory.POIS . . . 24
print.Trajectory.ZIP . . . 25
propAssign . . . 26
trajeR . . . 26
trajeR.BETA . . . 30
trajeR.CNORM . . . 32
trajeR.LOGIT . . . 35
trajeR.NL . . . 37
trajeR.POIS . . . 40
trajeR.ZIP . . . 42
trajeR_mult . . . 45
trajeRAIC . . . 46
trajeRBIC . . . 47

trajeRSH 47

Index **49**

---

adequacy *Model Adequacy Assessment*

Description

Provides a summary table of model adequacy metrics, including estimated group probabilities, confidence intervals, assignment proportions, Average Posterior Probability (AvePP), and Odds of Correct Classification (OCC).

Usage

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

Arguments

`sol`	A trajectory object returned by the `trajeR` function.
`Y`	The response variable matrix, as used in the 'trajeR' call.
`A`	The time variable matrix, as used in the 'trajeR' call.
`nb`	Integer. The number of repetitions for the bootstrap confidence interval.
`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 = "L")
adequacy(sol, Y = data[, 2:6], A = data[, 7:11])
```

AvePP *Average Posterior Probability of Assignment*

Description

The Average Posterior Probability (AvePP) measures the classification quality. For each group, it is the average of the highest posterior probabilities for all individuals assigned to that group. A value close to 1 indicates a clear classification.

Usage

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

Arguments

sol	A trajectory object returned by the trajeR function.
Y	The response variable matrix, as used in the 'trajeR' call.
A	The time variable matrix, as used in the 'trajeR' call.
X	Optional. A matrix of covariates that modify the group membership probability, if they were used in the model.

Value

A named numeric vector of the same length as the number of groups, containing the AvePP for each group.

Examples

```
data <- read.csv(system.file("extdata", "CNORM2gr.csv", package = "trajeR"))
data <- as.matrix(data)
# Use drop=FALSE for single-column matrices to preserve dimensions
sol <- trajeR(Y = data[, 2:6], A = data[, 7:11], Risk = data[, 12, drop = FALSE],
              degre = c(2, 2), Model = "CNORM", Method = "EM")
AvePP(sol, Y = data[, 2:6], A = data[, 7:11])
```

confidenceInt *Compute Confidence Intervals for Predicted Trajectories*

Description

Calculates predicted values and confidence intervals for the mean trajectory of each group at specified time points.

Usage

```
confidenceInt(Obj, newtime, alpha = 0.05)
```

Arguments

Obj	An object containing model parameters, including: • degre: Vector of polynomial degrees for each group. • varcov: Variance-covariance matrix of the model coefficients. • groups: Number of groups in the model. • Model: Character string specifying the model type (e.g., "CNORM", "LOGIT"). • beta: Vector of model coefficients.
newtime	Numeric vector of time points at which to compute predictions and confidence intervals.
alpha	The significance level for the confidence intervals. Default is 0.05, which corresponds to 95% confidence intervals.

Value

A list of class "Trajectory.predict" containing:

- newtime: The input time points.
- Ypred: Matrix of predicted values for each group at each time point.
- SEYpred: Matrix of standard errors for the predicted values.
- IC.inf: Matrix of the lower bounds of the confidence intervals.
- IC.sup: Matrix of the upper bounds of the confidence intervals.
- groups: Number of groups from the input object.

Examples

```
data <- read.csv(system.file("extdata", "CNORM2gr.csv", package = "trajeR"))
sol <- trajeR(Y = data[, 2:6], A = data[, 7:11], degre = c(2, 2), Model = "CNORM", Method = "L")
newtime <- seq(0, 1, by = 0.1)
result <- confidenceInt(sol, newtime, alpha = 0.05)
print(result)
```

ConfIntT *Bootstrap Confidence Intervals for Group Membership Probabilities*

Description

Calculates confidence intervals for the estimated group membership probabilities using a parametric bootstrap method. It simulates new theta parameters from a normal distribution based on the estimated theta and their standard errors.

Usage

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

Arguments

sol	A trajectory object returned by the trajeR function.
Y	The response variable matrix, as used in the 'trajeR' call.
A	The time variable matrix, as used in the 'trajeR' call.
nb	An integer. The number of repetitions in the bootstrap method.
alpha	A number. The significance level for the confidence interval (e.g., 0.05 for a 95% CI).

Value

A matrix with two rows (lower and upper bounds) and a column for each group, containing the confidence intervals for the estimated probabilities.

Examples

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

---

data_CNORM *Simulated CNORM Dataset*

Description

This dataset is a simulated dataset for use with the trajeR.CNORM function, which is an internal function to fit Censored Normal (CNORM) trajectory models. This dataset is likely used to demonstrate the functionality of the trajeR.CNORM model without specific censoring applied, or where censoring is handled implicitly by the model parameters rather than explicitly in the data.

The CNORM model is suitable for continuous outcome variables.

Usage

```
data_CNORM
```

Format

Expected to be a data format compatible with the trajeR.CNORM function, typically matrices for the response variable (Y), time variable (A), and potentially covariates (X, TCOV).

Source

Simulated data for illustrative purposes of trajectory analysis using the trajeR.CNORM model.

data_CNORM_Censored *Simulated Censored CNORM Dataset*

Description

This dataset is a simulated dataset specifically designed for use with the `trajeR.CNORM` function, demonstrating its capabilities with censored data. The `trajeR.CNORM` function is an internal function to fit Censored Normal (CNORM) models.

Censoring refers to observations where the true value is not precisely known but is known to be above or below a certain threshold. The `trajeR.CNORM` function supports both `ymin` (minimum value for censoring) and `ymax` (maximum value for censoring).

This dataset is expected to contain a response variable (Y), time variable (A), and potentially covariates (X, TCOV) suitable for trajectory analysis using the CNORM model, where some observations in Y are censored.

Usage

```
data_CNORM_Censored
```

Format

Expected to be a data format compatible with the `trajeR.CNORM` function, typically matrices for Y, A, X, and TCOV. The specific structure would be similar to other datasets used with `trajeR.CNORM`.

Source

Simulated data for illustrative purposes of censored trajectory analysis using the `trajeR.CNORM` model.

data_LOGIT *Simulated LOGIT Dataset*

Description

This dataset is a simulated dataset for use with the `trajeR.LOGIT` function, which is an internal function to fit Logistic trajectory models. This dataset is likely used to demonstrate the functionality of the `trajeR.LOGIT` model.

The LOGIT model is suitable for binary or categorical outcome variables, often representing probabilities or classifications.

Usage

```
data_LOGIT
```

Format

Expected to be a data format compatible with the `trajeR.LOGIT` function, typically matrices for the response variable (Y), time variable (A), and potentially covariates (X, TCOV).

Source

Simulated data for illustrative purposes of trajectory analysis using the `trajeR.LOGIT` model.

---

data_ZIP *Simulated ZIP Dataset*

Description

This dataset is a simulated dataset for use with the `trajeR.ZIP` function, which is an internal function to fit Zero-Inflated Poisson trajectory models. This dataset is likely used to demonstrate the functionality of the `trajeR.ZIP` model.

The ZIP model is appropriate for count data that has an excess of zero observations, often found in fields like public health or ecology.

Usage

```
data_ZIP
```

Format

Expected to be a data format compatible with the `trajeR.ZIP` function, typically matrices for the response variable (Y), time variable (A), and potentially covariates (X, TCOV).

Source

Simulated data for illustrative purposes of trajectory analysis using the `trajeR.ZIP` model.

---

dataNORM01 *Simulated CNORM Dataset (data01)*

Description

The `data01` dataset is a simulated dataset included in the package. It contains 500 trajectories with 10 time-points each, simulating a 3-group solution. The parameters for the groups are summarized in the table below:

Parameter	Group 1	Group 2	Group 3
β_{k0}	2.797	7.000	19.545
β_{k1}	8.809	-0.297	-0.407
β_{k2}	-3.201	0.463	0.026

β_{k3}	0.000	-0.021	0.000
π_k	0.32	0.54	0.14
σ_k	4.000	4.000	4.000

Usage

```
dataNORM01
```

Format

A data frame with 500 rows and 43 columns:

Columns 2:11 Matrix of real numbers (variable of interest, Y).

Columns 12:21 Matrix of time points (1 to 10, A).

Columns 22:31 Matrix of binary values (0 or 1, time-dependent covariate W).

Columns 32:41 Matrix of real numbers (additional time-dependent covariate W).

Columns 42:43 Matrix of binary values (0 or 1, covariate X).

Details

The dataset includes the following variables:

- `Y` (columns 2:11): Matrix of real numbers representing the variable of interest.
- `A` (columns 12:21): Matrix of time points (1 to 10).
- `W` (columns 22:31): Matrix with values 0 and 1, indicating the presence or absence of a characteristic.
- `W` (columns 32:41): Matrix of real numbers (additional time-dependent covariate).
- `X` (columns 42:43): Matrix with values 0 and 1, influencing group membership probability.

Source

Simulated data for illustrative purposes.

`diffaitbeta` *Differential*

Description

Differential

Usage

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

Arguments

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

Value

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

---

fait *Function fait*

Description

Function fait

Usage

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

Arguments

betak	Vector of integer.
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.

get_group_probabilities *Compute Posterior Probabilities and Group Assignments for Multivariate Trajectory Models*

Description

Calculates the posterior probabilities of group membership for each individual based on a fitted multivariate trajectory model. It assigns each individual to the most likely joint trajectory group and provides the specific sub-group assignments across all dimensions.

Usage

```
get_group_probabilities(
  object,
  lY,
  lA,
  Risk = NULL,
  lymin = NULL,
  lymax = NULL,
  lTCOV = NULL,
  ldelta = NULL,
  lnw = NULL
)
```

Arguments

object	An object of class `trajectory.mult` returned by the `trajeR_mult` function.
lY	List of matrices representing the longitudinal responses (one matrix per model). Must match the data used for fitting.
lA	List of matrices representing the time or age of observation (one matrix per model). Must match the data used for fitting.
Risk	An optional one-sided formula for global risk factors, matching the one used for fitting.
lymin	An optional list of minimum boundaries for censoring. If `NULL`, it is calculated automatically.
lymax	An optional list of maximum boundaries for censoring. If `NULL`, it is calculated automatically.
lTCOV	**(Not implemented)** List of time-dependent covariates.
ldelta	**(Not implemented)** List of matrices indicating the presence of time-dependent covariates.
lnw	**(Not implemented)** List of integers representing weights.

Value

A list containing the following components:

post_prob	A matrix of posterior probabilities for each individual and each joint group.
joint_group	A numeric vector indicating the most likely joint group for each individual.
group_assignments	A matrix showing the sub-group assignment for each joint group.

GroupProb *Calculate Posterior Membership Probabilities*

Description

GroupProb calculates the posterior probability for each individual to belong to each of the latent groups, based on a fitted trajeR model.

Usage

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

Arguments

Obj	A trajectory object returned by the trajeR function.
Y	Matrix. A real matrix. The data.
A	Matrix. A real matrix. The time variable.
TCOV	Matrix. Optional. A matrix containing the time-dependent covariates, if they were used in the model.
X	Matrix. Optional. A matrix of covariates that modify the group membership probability, if they were used in the model.

Value

A matrix of size n x ng (number of individuals by number of groups), where each entry [i, j] is the posterior probability of individual i belonging to group j.

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 *Generate Group Profiles Based on Covariates*

Description

This function calculates the mean of specified covariates for each trajectory group. It first assigns each individual to a group based on the highest posterior probability, then computes the average value of each provided covariate for all members of that group.

Usage

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

Arguments

sol	A trajectory object returned by the trajeR function.
Y	The response variable matrix, as used in the 'trajeR' call.
A	The time variable matrix, as used in the 'trajeR' call.
X	A numeric matrix or data frame of covariates for which the profiles are to be calculated. Each column represents a different covariate.

Value

A matrix where rows correspond to the covariates in 'X' and columns correspond to the trajectory groups. Each cell contains the mean of a covariate for a specific 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], Risk = data[, 12, drop = FALSE],
  degre = c(2, 2), Model = "CNORM", Method = "L"
)
GroupProfiles(sol, Y = data[, 2:6], A = data[, 7:11], X = data[, 12, drop = FALSE])
```

OCC *Odds of Correct Classification (OCC)*

Description

The Odds of Correct Classification (OCC) compares the odds of correct assignment using the model's posterior probabilities against the odds of assignment based on the estimated group proportions alone. A high OCC value suggests that the model provides a much better classification than a random assignment.

Usage

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

Arguments

sol	A trajectory object returned by the trajeR function.
Y	The response variable matrix, as used in the 'trajeR' call.
A	The time variable matrix, as used in the 'trajeR' call.

Value

A named numeric vector of the same length as the number of groups, containing the OCC for 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], Risk = data[, 12, drop = FALSE],
              degre = c(2, 2), Model = "CNORM", Method = "EM")
OCC(sol, Y = data[, 2:6], A = data[, 7:11])
```

---

plot_trajectory_mult *Plot Multivariate Trajectories*

Description

Plots a grid of longitudinal trajectories for a multivariate model. The grid displays each outcome (rows) and each latent group (columns). Individual trajectories are colored by their assigned group and faded, with the specific group highlighted and its average polynomial curve overlaid.

Usage

```
plot_trajectory_mult(object, lY, lA, assignments, base_colors = NULL)
```

Arguments

object	An object of class trajectory.mult.
lY	List of matrices representing the longitudinal responses.
lA	List of matrices representing the time or age of observation.
assignments	A matrix of group assignments (e.g., the group_assignments output from get_group_probabilities).
base_colors	Optional vector of hex colors for the groups.

plotrajeR *Plot Group Trajectories from a trajeR Model*

Description

A generic plotting function for objects of class 'Trajectory.*'. It visualizes both individual and average group trajectories.

Usage

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

Arguments

Obj	A trajectory object returned by the trajeR function.
...	Other parameters passed to the specific plot methods.

Value

This function is called for its side effect of creating a plot. It does not return any value.

Examples

```
# See examples for specific methods like plotrajeR.Trajectory.CNORM
data <- read.csv(system.file("extdata", "CNORM2gr.csv", package = "trajeR"))
sol <- trajeR(Y = data[, 2:6], A = data[, 7:11], Risk = data[, 12, drop = FALSE],
              degre = c(2, 2), Model = "CNORM", Method = "EM")
plotrajeR(sol)
```

plotrajeR.Trajectory.BETA *Plot Method for BETA Trajectory Objects*

Description

Plots the trajectories for a Beta model fitted by 'trajeR'.

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.BETA'.
plotcov	An optional vector or matrix of time-dependent covariates.
col	An optional vector of colors.
Y	The original response variable matrix.
A	The original time variable matrix.
Risk	Optional. The original risk covariate matrix.
TCOV	Matrix or Formula. An optional matrix (or one-sided formula) containing time-dependent covariates that influence the trajectories.
mean	Logical. If 'TRUE', plots the mean trajectory for each group.
alpha	A numeric value for the transparency of individual trajectory lines.
...	Other graphical parameters passed to 'plot'.

Value

a graphic.

---

plotrajeR.Trajectory.CNORM

Plot Method for CNORM Trajectory Objects

Description

Plots the trajectories for a Censored Normal model fitted by 'trajeR'.

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 optional vector or matrix of time-dependent covariates to visualize their effect on trajectories.
col	An optional vector of colors. If its length is '2 * ng', the first 'ng' colors are for individual trajectories and the next 'ng' are for the mean trajectories.
Y	The original response variable matrix used to fit the model. If provided, individual trajectories are plotted.
A	The original time variable matrix used to fit the model.
Risk	Optional. The original risk covariate matrix used to fit the model.
mean	Logical. If 'TRUE', the mean trajectory for each group (based on assigned individuals) is plotted as a dashed line.
alpha	A numeric value between 0 and 1 for the transparency of individual trajectory lines.
...	Other graphical parameters passed to 'plot'.

Value

a graphic.

---

plotrajeR.Trajectory.LOGIT

Plot Method for LOGIT Trajectory Objects

Description

Plots the trajectories for a Logistic model fitted by 'trajeR'.

Usage

```
## S3 method for class 'Trajectory.LOGIT'
plotrajeR(
  Obj,
  plotcov = NULL,
  dec = 0,
  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 of time-dependent covariates.
dec	A numeric value controlling the amount of jitter applied to data points for better visualization.
col	An optional vector of colors for individual points and mean trajectories.
Y	The original response variable matrix. If provided, individual data points are jittered and plotted.
A	The original time variable matrix.
Risk	Optional. The original risk covariate matrix.
mean	Logical. If 'TRUE', plots the mean of the observed data for each group.
alpha	A numeric value for the transparency of individual trajectory lines.
...	Other graphical parameters passed to 'plot'.

Value

a graphic.

---

plotrajeR.Trajectory.NL

Plot Method for Non-Linear Trajectory Objects

Description

Plots the trajectories for a non-linear model fitted by 'trajeR'.

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.NL'.
plotcov	An optional vector or matrix of time-dependent covariates.
col	An optional vector of colors.
Y	The original response variable matrix.
A	The original time variable matrix.
Risk	Optional. The original risk covariate matrix.
mean	Logical. If 'TRUE', plots the mean trajectory for each group.
alpha	A numeric value for the transparency of individual trajectory lines.
TCOV	Matrix or Formula. An optional matrix (or one-sided formula) containing time-dependent covariates that influence the trajectories.
...	Other graphical parameters passed to 'plot'.

Value

a graphic.

---

plotrajeR.Trajectory.POIS

Plot Method for POIS Trajectory Objects

Description

Plots the trajectories for a Poisson model fitted by 'trajeR'.

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 of time-dependent covariates.
dec	A numeric value controlling the amount of jitter applied to data points.
col	An optional vector of colors.
Y	The original response variable matrix. If provided, individual trajectories are plotted.
A	The original time variable matrix.
Risk	Optional. The original risk covariate matrix.
TCOV	Matrix or Formula. An optional matrix (or one-sided formula) containing time-dependent covariates that influence the trajectories.
mean	Logical. If 'TRUE', plots the mean trajectory for each group.
alpha	A numeric value for the transparency of individual trajectory lines.
...	Other graphical parameters passed to 'plot'.

Value

a graphic.

---

plotrajeR.Trajectory.ZIP

Plot Method for ZIP Trajectory Objects

Description

Plots the trajectories for a Zero-Inflated Poisson (ZIP) model fitted by 'trajeR'.

Usage

```
## S3 method for class 'Trajectory.ZIP'
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.ZIP'.
plotcov	An optional vector or matrix of time-dependent covariates.
dec	A numeric value controlling the amount of jitter applied to data points.
col	An optional vector of colors.
Y	The original response variable matrix.
A	The original time variable matrix.
Risk	Optional. The original risk covariate matrix.
TCOV	Matrix or Formula. An optional matrix (or one-sided formula) containing time-dependent covariates that influence the trajectories.
mean	Logical. If 'TRUE', plots the mean trajectory for each group.
alpha	A numeric value for the transparency of individual trajectory lines.
...	Other graphical parameters passed to 'plot'.

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 Method for a 'Trajectory.CNORM' Object

Description

Displays a summary of a censored normal trajectory model fitted by 'trajeR'.

Usage

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

Arguments

x	An object of class 'Trajectory.CNORM'.
...	Further arguments passed to or from other methods (not currently used).

Value

Invisibly returns the original object 'x'.

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")
sol
```

`print.Trajectory.LOGIT` *Print Method for a 'Trajectory.LOGIT' Object*

Description

Displays a summary of a logistic trajectory model fitted by 'trajeR'.

Usage

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

Arguments

`x`	An object of class 'Trajectory.LOGIT'.
`...`	Further arguments passed to or from other methods (not currently used).

Value

Invisibly returns the original object 'x'.

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], degre = c(1, 2), Model = "LOGIT", Method = "L")
sol
```

`print.trajectory.mult` *Print Method for a 'trajectory.mult' Object*

Description

Displays a summary of a multivariate (joint) trajectory model fitted by 'trajeR_mult'.

Usage

```
## S3 method for class 'trajectory.mult'
print(x, ...)
```

Arguments

x	An object of class ‘trajectory.mult‘.
...	Further arguments passed to or from other methods.

Value

Invisibly returns the original object ‘x‘.

---

print.Trajectory.NL *Print Method for a ‘Trajectory.NL‘ Object*

Description

Displays a summary of a non-linear trajectory model fitted by ‘trajeR‘.

Usage

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

Arguments

x	An object of class ‘Trajectory.NL‘.
...	Further arguments passed to or from other methods (not currently used).

Value

Invisibly returns the original object ‘x‘.

---

print.Trajectory.POIS *Print Method for a ‘Trajectory.ZIP‘ Object*

Description

Displays a summary of a zero-inflated Poisson trajectory model fitted by ‘trajeR‘.

Usage

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

Arguments

x	An object of class ‘Trajectory.ZIP‘.
...	Further arguments passed to or from other methods (not currently used).

Value

Invisibly returns the original object ‘x‘.

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 *Proportion of Individuals Assigned to Each Group*

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	A trajectory object returned by the trajeR function.
Y	The response variable matrix, as used in the 'trajeR' call.
A	The time variable matrix, as used in the 'trajeR' call.

Value

A named numeric vector containing the proportion of individuals assigned to 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], Risk = data[, 12, drop = FALSE],
              degre = c(2, 2), Model = "CNORM", Method = "EM")
propAssign(sol, Y = data[, 2:6], A = data[, 7:11])
```

trajeR *Fitting longitudinal mixture models*

Description

Fitting longitudinal mixture models

Usage

```
trajeR(
  Y,
  A = NULL,
  Risk = NULL,
  TCOV = NULL,
  data = NULL,
  ng = NULL,
```

```
  degre = NULL,
  degre.nu = 0,
  degre.phi = 0,
  Model,
  Method = "L",
  ssigma = FALSE,
  ymax = NULL,
  ymin = NULL,
  hessian = TRUE,
  itermax = 100,
  paraminit = NULL,
  ProbIRLS = TRUE,
  refgr = 1,
  fct = NULL,
  diffct = NULL,
  nbvar = NULL,
  ng.nl = NULL,
  nls.lmiter = 50,
  control = list()
)
```

Arguments

Y	Matrix, Formula, or a list of Formulas. A matrix containing the response variables, a single formula (e.g., `Y ~ A + I(A^2)`), or a list of formulas for models with different degrees.
A	Matrix. A matrix containing the time variable data. Can be omitted if specified in the RHS of the `Y` formula.
Risk	Matrix or Formula. An optional matrix (or one-sided formula like `~ X1 + X2`) of covariates that modify the probability of belonging to a group.
TCOV	Matrix or Formula. An optional matrix (or one-sided formula) containing time-dependent covariates that influence the trajectories.
data	data.frame. An optional data frame containing the variables named in the formulas.
ng	Integer. The number of groups. Required if using a single formula or if 'degre' is not provided.
degre	Vector of integers. The degree of each polynomial function. The code adds 1 to these values internally.
degre.nu	Vector of integers. The degree of the polynomial for the zero-inflation part of a ZIP model.
degre.phi	Vector of integers. The degree of the polynomial for the precision parameter (phi) of a BETA model. The code adds 1 to these values internally.
Model	String. The model to be used. One of "LOGIT", "CNORM", "ZIP", "BETA", "POIS", or a non-linear model.
Method	String. The estimation method. "L" for Likelihood, "EM" for Expectation-Maximization.

ssigma	Logical. For the CNORM model, if TRUE, a single shared standard deviation (sigma) is estimated for all groups. Default is FALSE.
ymax	Real. For the CNORM model, the maximum value for censoring. Defaults to max(Y) + 1.
ymin	Real. For the CNORM model, the minimum value for censoring. Defaults to min(Y) - 1.
hessian	Logical. If TRUE, the Hessian matrix is computed. Default is TRUE.
itermax	Integer. The maximum number of iterations for the optimization algorithm (like 'optim' or the EM loop). Default is 100.
paraminit	Vector. The vector of initial parameters.
ProbIRLS	Logical. Indicates the method to use for searching predictor probabilities. Default is TRUE.
refgr	Integer. The reference group number. 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.
nbvar	Integer. The number of variable in the nonlinear model.
ng.nl	Integer. The number of group for a non linear model.
nls.lmiter	Integer. In the case of non linear model, the maximum number of iterations allowed.
control	A list of control parameters for the 'ucminf' optimization algorithm.

- 'trace': If positive, detailed tracing information is printed for each iteration.
- 'grtol': The algorithm stops when the largest absolute value of the gradient is less than 'grtol'. Default is 1e-6.
- 'xtol': The algorithm stops when the last relative step length is sufficiently small. Default is 1e-12.
- 'stepmax': Initial maximal allowed step length (radius of trust-region). The value is updated during the optimization. Default is 1.
- 'maxeval': The maximum number of function evaluations (objective and gradient). Default is 500.
- 'grad': Method for finite difference approximation of the gradient if not provided. Can be 'forward' (default) or 'central'.
- 'gradstep': Vector of length 2 for the step length in finite difference approximation. Default is c(1e-6, 1e-8).

Value

Returns an object of a class corresponding to the chosen model (e.g., 'Trajectory.CNORM', 'Trajectory.LOGIT').

Returns an object of a class corresponding to the chosen model (e.g., 'Trajectory.CNORM', 'Trajectory.LOGIT'). This object, referred to as a 'Trajectory' object, contains the following elements:

- 'beta': Vector of the final beta parameters for the trajectory shapes.
- 'sigma': Vector of the final sigma parameters (standard deviations), for CNORM and NL models.

- ‘nu‘: Vector of final nu parameters for the zero-inflation part of a ZIP model.
- ‘phi‘: Vector of final phi parameters (precision) for a BETA model.
- ‘delta‘: Vector of the final delta parameters for time-dependent covariates (if any).
- ‘theta‘: Vector of the final theta parameters for group membership probabilities.
- ‘sd‘: Vector of the standard deviations of the estimated parameters.
- ‘tab‘: A data frame with parameter estimates, standard errors, T-values, and p-values.
- ‘Model‘: A string indicating the fitted model (e.g., "CNORM", "LOGIT").
- ‘groups‘: The number of groups.
- ‘Names‘: The names of the parameters.
- ‘Method‘: The estimation method used ("L" for Likelihood or "EM" for Expectation-Maximization).
- ‘Size‘: The number of individuals (observations).
- ‘Likelihood‘: The final log-likelihood value.
- ‘Time‘: The time points for the first individual, or min/max time for ZIP model.
- ‘degre‘: A vector with the polynomial degrees for each group’s trajectory.
- ‘degre.nu‘: For ZIP models, a vector with the polynomial degrees for the zero-inflation part.
- ‘degre.phi‘: For BETA models, a vector with the polynomial degrees for the precision part.
- ‘min‘, ‘max‘: For CNORM models, the censoring limits used.
- ‘fct‘: For NL models, the non-linear function provided.
- ‘varcov‘: The variance-covariance matrix of the parameters.
- ‘convergence‘: An integer code indicating the convergence status of the optimization algorithm. The meaning depends on the method used.
  - For ‘Method = "L"‘ (using ‘ucminf‘):
  - ‘1‘: Stopped by small gradient (‘grtol‘). The current iterate is probably a solution.
  - ‘2‘: Stopped by small step (‘xtol‘). Successive iterates are within tolerance.
  - ‘3‘: Stopped by function evaluation limit (‘maxeval‘).
  - ‘4‘: Stopped by zero step from line search. May be a solution.
  - ‘-2‘: Computation did not start: length of parameters is 0.
  - ‘-4‘: Computation did not start: ‘stepmax‘ is too small.
  - ‘-5‘: Computation did not start: ‘grtol‘ or ‘xtol‘ is not positive.
  - ‘-6‘: Computation did not start: ‘maxeval‘ is not positive.
  - ‘-7‘: Computation did not start: The provided Hessian was not positive definite.
  - For ‘Method = "EM"‘:
  - ‘1‘: Indicates that the EM algorithm has completed its run (convergence is assessed by the user by checking the stability of parameters across iterations).

`trajeR.BETA` *Internal function to fit Beta Regression Model*

Description

Internal function to fit Beta Regression Model

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,
  control
)
```

Arguments

`Y`	Matrix, Formula, or a list of Formulas. A matrix containing the response variables, a single formula (e.g., `Y ~ A + I(A^2)`), or a list of formulas for models with different degrees.
`A`	Matrix. A matrix containing the time variable data. Can be omitted if specified in the RHS of the `Y` formula.
`X`	Matrix. The matrix of covariates modifying group membership probability. Includes an intercept.

TCOV	Matrix or Formula. An optional matrix (or one-sided formula) containing time-dependent covariates that influence the trajectories.
ng	Integer. The number of groups. Required if using a single formula or if ‘degre‘ is not provided.
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 covariates.
ntheta	Integer. Number of theta parameters for each group.
period	Integer. Number of measurement time points.
degre	Vector of integers. The degree of each polynomial function. The code adds 1 to these values internally.
theta	Vector of real. Initial parameters for group membership probability.
beta	List of vectors. Initial beta parameters for each group's trajectory.
phi	List of vectors. Initial phi (precision) parameters.
delta	List of vectors. Initial delta parameters for time-dependent covariates.
pi	Vector of real. Initial group membership probabilities.
Method	String. The estimation method. "L" for Likelihood, "EM" for Expectation-Maximization.
hessian	Logical. If TRUE, the Hessian matrix is computed. Default is TRUE.
itermax	Integer. The maximum number of iterations for the optimization algorithm (like ‘optim‘ or the EM loop). Default is 100.
paraminit	Vector. The vector of initial parameters.
EMIRLS	Logical. If TRUE, use the EM-IRLS method for estimation.
refgr	Integer. The reference group number. Default is 1.
control	A list of control parameters for the ‘ucminf‘ optimization algorithm. • ‘trace‘: If positive, detailed tracing information is printed for each iteration. • ‘grtol‘: The algorithm stops when the largest absolute value of the gradient is less than ‘grtol‘. Default is 1e-6. • ‘xtol‘: The algorithm stops when the last relative step length is sufficiently small. Default is 1e-12. • ‘stepmax‘: Initial maximal allowed step length (radius of trust-region). The value is updated during the optimization. Default is 1. • ‘maxeval‘: The maximum number of function evaluations (objective and gradient). Default is 500. • ‘grad‘: Method for finite difference approximation of the gradient if not provided. Can be 'forward' (default) or 'central'. • ‘gradstep‘: Vector of length 2 for the step length in finite difference approximation. Default is c(1e-6, 1e-8).

Value

Returns an object of class ‘Trajectory.BETA‘.

- ‘beta‘: Vector of final beta parameters (mean).
- ‘phi‘: Vector of final phi parameters (precision).
- ‘delta‘: Vector of final delta parameters (if ‘nw > 0‘).
- ‘theta‘: Vector of final theta parameters for group membership.
- ‘sd‘: Vector of standard deviations of the parameters.
- ‘tab‘: A data frame with parameter estimates and statistics.
- ‘Model‘: The string "BETA".
- ‘groups‘: The number of groups.
- ‘Names‘: The names of the parameters.
- ‘Method‘: The estimation method used.
- ‘Size‘: The number of individuals.
- ‘Likelihood‘: The final log-likelihood value.
- ‘Time‘: The time points for the first individual.
- ‘degre‘: Polynomial degrees for the mean part.
- ‘degre.phi‘: Polynomial degrees for the precision part.

---

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,
  ssigma,
  ymax,
  ymin,
  hessian,
  itermax,
  paraminit,
  EMIRLS,
  refgr,
  control
)
```

Arguments

Y	Matrix, Formula, or a list of Formulas. A matrix containing the response variables, a single formula (e.g., `Y ~ A + I(A^2)`), or a list of formulas for models with different degrees.
A	Matrix. A matrix containing the time variable data. Can be omitted if specified in the RHS of the `Y` formula.
X	Matrix. The matrix of covariates modifying group membership probability. Includes an intercept.
TCOV	Matrix or Formula. An optional matrix (or one-sided formula) containing time-dependent covariates that influence the trajectories.
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 covariates.
ntheta	Integer. Number of theta parameters for each group.
period	Integer. Number of measurement time points.
degre	Vector of integers. The degree of each polynomial function. The code adds 1 to these values internally.
theta	Vector of real. Initial parameters for group membership probability.
beta	List of vectors. Initial beta parameters for each group's trajectory.
sigma	Vector of real. Initial sigma parameters (standard deviations) for each group.
delta	List of vectors. Initial delta parameters for time-dependent covariates.
pi	Vector of real. Initial group membership probabilities.
Method	String. The estimation method. "L" for Likelihood, "EM" for Expectation-Maximization.

ssigma	Logical. For the CNORM model, if TRUE, a single shared standard deviation (sigma) is estimated for all groups. Default is FALSE.
ymax	Real. For the CNORM model, the maximum value for censoring. Defaults to max(Y) + 1.
ymin	Real. For the CNORM model, the minimum value for censoring. Defaults to min(Y) - 1.
hessian	Logical. If TRUE, the Hessian matrix is computed. Default is TRUE.
itermax	Integer. The maximum number of iterations for the optimization algorithm (like 'optim' or the EM loop). Default is 100.
paraminit	Vector. The vector of initial parameters.
EMIRLS	Logical. If TRUE, use the EM-IRLS method for estimation.
refgr	Integer. The reference group number. Default is 1.
control	A list of control parameters for the 'ucminf' optimization algorithm.

- 'trace': If positive, detailed tracing information is printed for each iteration.
- 'grtol': The algorithm stops when the largest absolute value of the gradient is less than 'grtol'. Default is 1e-6.
- 'xtol': The algorithm stops when the last relative step length is sufficiently small. Default is 1e-12.
- 'stepmax': Initial maximal allowed step length (radius of trust-region). The value is updated during the optimization. Default is 1.
- 'maxeval': The maximum number of function evaluations (objective and gradient). Default is 500.
- 'grad': Method for finite difference approximation of the gradient if not provided. Can be 'forward' (default) or 'central'.
- 'gradstep': Vector of length 2 for the step length in finite difference approximation. Default is c(1e-6, 1e-8).

Value

Returns an object of class 'Trajectory.CNORM'.

- 'beta': Vector of the final beta parameters.
- 'sigma': Vector of the final sigma parameters.
- 'delta': Vector of the final delta parameters (if 'nw > 0').
- 'theta': Vector of the final theta parameters for group membership.
- 'sd': Vector of the standard deviations of the parameters.
- 'tab': A data frame with parameter estimates, standard errors, T-values, and p-values.
- 'Model': The string "CNORM".
- 'groups': The number of groups.
- 'Names': The names of the parameters.
- 'Method': The estimation method used ("L" or "EM").
- 'Size': The number of individuals.

- 'Likelihood': The final log-likelihood value.
- 'Time': The time points for the first individual.
- 'degre': A vector with the polynomial degrees for each group.
- 'min', 'max': The censoring limits used.

`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,
  control
)
```

Arguments

`Y`	Matrix, Formula, or a list of Formulas. A matrix containing the response variables, a single formula (e.g., `Y ~ A + I(A^2)`), or a list of formulas for models with different degrees.

A	Matrix. A matrix containing the time variable data. Can be omitted if specified in the RHS of the Y formula.
X	Matrix. The matrix of covariates modifying group membership probability. Includes an intercept.
TCOV	Matrix or Formula. An optional matrix (or one-sided formula) containing time-dependent covariates that influence the trajectories.
ng	Integer. The number of groups. Required if using a single formula or if 'degre' is not provided.
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 covariates.
ntheta	Integer. Number of theta parameters for each group.
period	Integer. Number of measurement time points.
degre	Vector of integers. The degree of each polynomial function. The code adds 1 to these values internally.
theta	Vector of real. Initial parameters for group membership probability.
beta	List of vectors. Initial beta parameters for each group's trajectory.
delta	List of vectors. Initial delta parameters for time-dependent covariates.
pi	Vector of real. Initial group membership probabilities.
Method	String. The estimation method. "L" for Likelihood, "EM" for Expectation-Maximization.
hessian	Logical. If TRUE, the Hessian matrix is computed. Default is TRUE.
itermax	Integer. The maximum number of iterations for the optimization algorithm (like 'optim' or the EM loop). Default is 100.
paraminit	Vector. The vector of initial parameters.
EMIRLS	Logical. If TRUE, use the EM-IRLS method for estimation.
refgr	Integer. The reference group number. Default is 1.
control	A list of control parameters for the 'ucminf' optimization algorithm. • 'trace': If positive, detailed tracing information is printed for each iteration. • 'grtol': The algorithm stops when the largest absolute value of the gradient is less than 'grtol'. Default is 1e-6. • 'xtol': The algorithm stops when the last relative step length is sufficiently small. Default is 1e-12. • 'stepmax': Initial maximal allowed step length (radius of trust-region). The value is updated during the optimization. Default is 1. • 'maxeval': The maximum number of function evaluations (objective and gradient). Default is 500. • 'grad': Method for finite difference approximation of the gradient if not provided. Can be 'forward' (default) or 'central'. • 'gradstep': Vector of length 2 for the step length in finite difference approximation. Default is c(1e-6, 1e-8).

Value

Returns an object of class ‘Trajectory.LOGIT‘.

- ‘beta‘: Vector of the final beta parameters.
- ‘delta‘: Vector of the final delta parameters (if ‘nw > 0‘).
- ‘theta‘: Vector of the final theta parameters for group membership.
- ‘sd‘: Vector of the standard deviations of the parameters.
- ‘tab‘: A data frame with parameter estimates, standard errors, T-values, and p-values.
- ‘Model‘: The string "LOGIT".
- ‘groups‘: The number of groups.
- ‘Names‘: The names of the parameters.
- ‘Method‘: The estimation method used.
- ‘Size‘: The number of individuals.
- ‘Likelihood‘: The final log-likelihood value.
- ‘Time‘: The time points for the first individual.
- ‘degre‘: A vector with the polynomial degrees for each group.

`trajeR.NL` *Internal function to fit a Non-Linear Model*

Description

Internal function to fit a 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.lmiter,
  control
)
```

Arguments

Y	Matrix, Formula, or a list of Formulas. A matrix containing the response variables, a single formula (e.g., `Y ~ A + I(A^2)`), or a list of formulas for models with different degrees.
A	Matrix. A matrix containing the time variable data. Can be omitted if specified in the RHS of the `Y` formula.
X	Matrix. The matrix of covariates modifying group membership probability. Includes an intercept.
TCOV	Matrix or Formula. An optional matrix (or one-sided formula) containing time-dependent covariates that influence the trajectories.
ng	Integer. The number of groups. Required if using a single formula or if 'degre' is not provided.
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 covariates.
ntheta	Integer. Number of theta parameters for each group.
period	Integer. Number of measurement time points.
degre	Vector of integers. The degree of each polynomial function. The code adds 1 to these values internally.
theta	Vector of real. Initial parameters for group membership probability.
beta	List of vectors. Initial beta parameters for each group's trajectory.
sigma	Vector of real. Initial sigma parameters (standard deviations) for each group.
pi	Vector of real. Initial group membership probabilities.
Method	String. The estimation method. "L" for Likelihood, "EM" for Expectation-Maximization.
ssigma	Logical. For the CNORM model, if TRUE, a single shared standard deviation (sigma) is estimated for all groups. Default is FALSE.
hessian	Logical. If TRUE, the Hessian matrix is computed. Default is TRUE.

`itermax`	Integer. The maximum number of iterations for the optimization algorithm (like 'optim' or the EM loop). Default is 100.
`paraminit`	Vector. The vector of initial parameters.
`EMIRLS`	Logical. If TRUE, use the EM-IRLS method for estimation.
`refgr`	Integer. The reference group number. 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.lmiter`	Integer. In the case of non linear model, the maximum number of iterations allowed.
`control`	A list of control parameters for the 'ucminf' optimization algorithm.

- 'trace': If positive, detailed tracing information is printed for each iteration.
- 'grtol': The algorithm stops when the largest absolute value of the gradient is less than 'grtol'. Default is 1e-6.
- 'xtol': The algorithm stops when the last relative step length is sufficiently small. Default is 1e-12.
- 'stepmax': Initial maximal allowed step length (radius of trust-region). The value is updated during the optimization. Default is 1.
- 'maxeval': The maximum number of function evaluations (objective and gradient). Default is 500.
- 'grad': Method for finite difference approximation of the gradient if not provided. Can be 'forward' (default) or 'central'.
- 'gradstep': Vector of length 2 for the step length in finite difference approximation. Default is c(1e-6, 1e-8).

Value

Returns an object of class 'Trajectory.NL'.

- 'beta': Vector of final beta parameters.
- 'sigma': Vector of final sigma parameters.
- 'delta': Vector of final delta parameters (if 'nw > 0').
- 'theta': Vector of final theta parameters for group membership.
- 'sd': Vector of standard deviations of the parameters.
- 'tab': A data frame with parameter estimates and statistics.
- 'Model': The string "CNORM" (Note: should probably be "NL").
- 'groups': The number of groups.
- 'Names': The names of the parameters.
- 'Method': The estimation method used.
- 'Size': The number of individuals.
- 'Likelihood': The final log-likelihood value.
- 'Time': The time points for the first individual.
- 'degre': A vector with the number of variables for the non-linear function.
- 'fct': The non-linear function provided.

`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,
  control
)
```

Arguments

`Y`	Matrix, Formula, or a list of Formulas. A matrix containing the response variables, a single formula (e.g., `Y ~ A + I(A^2)`), or a list of formulas for models with different degrees.
`A`	Matrix. A matrix containing the time variable data. Can be omitted if specified in the RHS of the `Y` formula.
`X`	Matrix. The matrix of covariates modifying group membership probability. Includes an intercept.
`TCOV`	Matrix or Formula. An optional matrix (or one-sided formula) containing time-dependent covariates that influence the trajectories.

ng	Integer. The number of groups. Required if using a single formula or if ‘degre‘ is not provided.
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 covariates.
ntheta	Integer. Number of theta parameters for each group.
period	Integer. Number of measurement time points.
degre	Vector of integers. The degree of each polynomial function. The code adds 1 to these values internally.
theta	Vector of real. Initial parameters for group membership probability.
beta	List of vectors. Initial beta parameters for each group’s trajectory.
delta	List of vectors. Initial delta parameters for time-dependent covariates.
pi	Vector of real. Initial group membership probabilities.
Method	String. The estimation method. "L" for Likelihood, "EM" for Expectation-Maximization.
hessian	Logical. If TRUE, the Hessian matrix is computed. Default is TRUE.
itermax	Integer. The maximum number of iterations for the optimization algorithm (like ‘optim‘ or the EM loop). Default is 100.
paraminit	Vector. The vector of initial parameters.
EMIRLS	Logical. If TRUE, use the EM-IRLS method for estimation.
refgr	Integer. The reference group number. Default is 1.
control	A list of control parameters for the ‘ucminf‘ optimization algorithm. • ‘trace‘: If positive, detailed tracing information is printed for each iteration. • ‘grtol‘: The algorithm stops when the largest absolute value of the gradient is less than ‘grtol‘. Default is 1e-6. • ‘xtol‘: The algorithm stops when the last relative step length is sufficiently small. Default is 1e-12. • ‘stepmax‘: Initial maximal allowed step length (radius of trust-region). The value is updated during the optimization. Default is 1. • ‘maxeval‘: The maximum number of function evaluations (objective and gradient). Default is 500. • ‘grad‘: Method for finite difference approximation of the gradient if not provided. Can be ’forward’ (default) or ’central’. • ‘gradstep‘: Vector of length 2 for the step length in finite difference approximation. Default is c(1e-6, 1e-8).

Value

Returns an object of class ‘Trajectory.POIS‘.

- ‘beta‘: Vector of final beta parameters.

- ‘delta‘: Vector of final delta parameters (if ‘nw > 0‘).
- ‘theta‘: Vector of final theta parameters for group membership.
- ‘sd‘: Vector of standard deviations of the parameters.
- ‘tab‘: A data frame with parameter estimates and statistics.
- ‘Model‘: The string "POIS".
- ‘groups‘: The number of groups.
- ‘Names‘: The names of the parameters.
- ‘Method‘: The estimation method used.
- ‘Size‘: The number of individuals.
- ‘Likelihood‘: The final log-likelihood value.
- ‘Time‘: The time points for the first individual.
- ‘degre‘: A vector with the polynomial degrees for each group.

`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,
  control
)
```

Arguments

Y	Matrix, Formula, or a list of Formulas. A matrix containing the response variables, a single formula (e.g., `Y ~ A + I(A^2)`), or a list of formulas for models with different degrees.
A	Matrix. A matrix containing the time variable data. Can be omitted if specified in the RHS of the `Y` formula.
X	Matrix. The matrix of covariates modifying group membership probability. Includes an intercept.
TCOV	Matrix or Formula. An optional matrix (or one-sided formula) containing time-dependent covariates that influence the trajectories.
ng	Integer. The number of groups. Required if using a single formula or if 'degre' is not provided.
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 covariates.
ntheta	Integer. Number of theta parameters for each group.
period	Integer. Number of measurement time points.
degre	Vector of integers. The degree of each polynomial function. The code adds 1 to these values internally.
degre.nu	Vector of integers. The degree of the polynomial for the zero-inflation part of a ZIP model.
theta	Vector of real. Initial parameters for group membership probability.
beta	List of vectors. Initial beta parameters for each group's trajectory.
nu	List of vectors. Initial nu parameters for the zero-inflation part.
delta	List of vectors. Initial delta parameters for time-dependent covariates.
pi	Vector of real. Initial group membership probabilities.
Method	String. The estimation method. "L" for Likelihood, "EM" for Expectation-Maximization.
hessian	Logical. If TRUE, the Hessian matrix is computed. Default is TRUE.
itermax	Integer. The maximum number of iterations for the optimization algorithm (like 'optim' or the EM loop). Default is 100.
paraminit	Vector. The vector of initial parameters.
EMIRLS	Logical. If TRUE, use the EM-IRLS method for estimation.

`refgr` Integer. The reference group number. Default is 1.

`control` A list of control parameters for the ‘ucminf‘ optimization algorithm.

- ‘trace‘: If positive, detailed tracing information is printed for each iteration.
- ‘grtol‘: The algorithm stops when the largest absolute value of the gradient is less than ‘grtol‘. Default is 1e-6.
- ‘xtol‘: The algorithm stops when the last relative step length is sufficiently small. Default is 1e-12.
- ‘stepmax‘: Initial maximal allowed step length (radius of trust-region). The value is updated during the optimization. Default is 1.
- ‘maxeval‘: The maximum number of function evaluations (objective and gradient). Default is 500.
- ‘grad‘: Method for finite difference approximation of the gradient if not provided. Can be ’forward’ (default) or ’central’.
- ‘gradstep‘: Vector of length 2 for the step length in finite difference approximation. Default is c(1e-6, 1e-8).

Value

Returns an object of class ‘Trajectory.ZIP‘.

- ‘beta‘: Vector of final beta parameters (Poisson part).
- ‘nu‘: Vector of final nu parameters (zero-inflation part).
- ‘delta‘: Vector of final delta parameters (if ‘nw > 0‘).
- ‘theta‘: Vector of final theta parameters for group membership.
- ‘sd‘: Vector of standard deviations of the parameters.
- ‘tab‘: A data frame with parameter estimates and statistics.
- ‘Model‘: The string "ZIP".
- ‘groups‘: The number of groups.
- ‘Names‘: The names of the parameters.
- ‘Method‘: The estimation method used.
- ‘Size‘: The number of individuals.
- ‘Likelihood‘: The final log-likelihood value.
- ‘Time‘: A vector with min and max time values.
- ‘degre‘: Polynomial degrees for the Poisson part.
- ‘degre.nu‘: Polynomial degrees for the zero-inflation part.

`trajeR_mult` *Fitting Multivariate Longitudinal Mixture Models*

Description

Fits a multivariate (joint) group-based trajectory model for multiple longitudinal outcomes.

Usage

```
trajeR_mult(
  formulas,
  ng,
  models,
  Method = "L",
  Risk = NULL,
  TCOV = NULL,
  data = NULL,
  degre.nu = NULL,
  degre.phi = NULL,
  ssigma = TRUE,
  ymax = NULL,
  ymin = NULL,
  paraminit = NULL,
  control = list(maxit = 100),
  hessian = FALSE
)
```

Arguments

`formulas`	A list of formulas, one for each longitudinal outcome (e.g., `list(Y1 ~ Time, Y2 ~ Time + I(Time^2))`).
`ng`	A list of integers specifying the number of latent groups for each outcome.
`models`	A character vector of distribution models for each outcome (e.g., c("CNORM", "LOGIT")). Supported models are "CNORM", "LOGIT", "ZIP", "BETA", "POIS".
`Method`	The estimation method. "L" for Likelihood or "EM" for Expectation-Maximization.
`Risk`	An optional one-sided formula for global risk factors that influence group membership (e.g., `~ X1 + X2`).
`TCOV`	Formula or Matrix. Optional time-dependent covariates.
`data`	data.frame. The dataset containing all variables.
`degre.nu`	A list or vector specifying the polynomial degree for the zero-inflation part of "ZIP" models.
`degre.phi`	A list or vector specifying the polynomial degree for the precision parameter (phi) of "BETA" models.

ssigma	Logical. If TRUE, assumes a common standard deviation across groups for "CNORM" models. Can be a vector of logicals, one for each CNORM model. Default is TRUE.
ymax	A list or vector of maximum values for censoring in "CNORM" models.
ymin	A list or vector of minimum values for censoring in "CNORM" models.
paraminit	An optional list of numeric vectors for user-defined initial parameters. The list should have one element per model, plus an optional last element for the psi parameters.
control	A list of control parameters for the optimization algorithm (e.g., `list(maxit=100)`).
hessian	Logical. If TRUE, computes the Hessian matrix to estimate standard errors. Default is FALSE.

Value

An object of class `trajectory.mult` containing the fitted model parameters for each outcome, the interaction parameters (psi), likelihood, and convergence status.

`trajeRAIC` *Akaike Information Criterion (AIC) for a Trajectory Object*

Description

Calculates the Akaike Information Criterion (AIC) for a model fitted by ‘trajeR‘.

Usage

```
trajeRAIC(sol)
```

Arguments

sol	A trajectory object returned by the ‘trajeR‘ function.

Value

A numeric value representing the AIC.

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 *Bayesian Information Criterion (BIC) for a Trajectory Object*

Description

Calculates the Bayesian Information Criterion (BIC) for a model fitted by ‘trajeR‘.

Usage

```
trajeRBIC(sol)
```

Arguments

sol	A trajectory object returned by the ‘trajeR‘ function.

Value

A numeric value representing the BIC.

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 *Slope Heuristic for Trajectory Model Selection*

Description

Applies the slope heuristic (from the ‘capushe‘ package) to a list of ‘trajeR‘ models to select the best one.

Usage

```
trajeRSH(l)
```

Arguments

l	A list of trajectory objects returned by the ‘trajeR‘ function.

Value

An object of class ‘DDSE‘ from the ‘capushe‘ package, containing the selected model and related information.

Examples

```
data <- read.csv(system.file("extdata", "CNORM2gr.csv", package = "trajeR"))
degre <- list(
 c(2, 2),
 c(1, 1),
 c(1, 2),
 c(2, 1),
 c(2, 0),
 c(0, 2),
 c(3, 2),
 c(3, 1),
 c(2, 3),
 c(1, 3)
)
sol <- list()
for (i in 1:length(degre)) {
  sol[[i]] <- trajeR(
    Y = data[, 2:6], A = data[, 7:11],
    degre = degre[[i]], Model = "CNORM", Method = "EM"
  )
}
trajeRSH(sol)
```

Index

∗ **datasets**
  data_CNORM, 6
  data_CNORM_Censored, 7
  data_LOGIT, 7
  data_ZIP, 8
  dataNORM01, 8

adequacy, 3
AvePP, 4

confidenceInt, 4
ConfIntT, 5

data_CNORM, 6
data_CNORM_Censored, 7
data_LOGIT, 7
data_ZIP, 8
dataNORM01, 8
diffaitbeta, 9

fait, 10

get_group_probabilities, 11
GroupProb, 12
GroupProfiles, 13

OCC, 13

plot_trajectory_mult, 14
plotrajeR, 15
plotrajeR.Trajectory.BETA, 15
plotrajeR.Trajectory.CNORM, 16
plotrajeR.Trajectory.LOGIT, 17
plotrajeR.Trajectory.NL, 18
plotrajeR.Trajectory.POIS, 19
plotrajeR.Trajectory.ZIP, 20
print.Trajectory.BETA, 21
print.Trajectory.CNORM, 22
print.Trajectory.LOGIT, 23
print.trajectory.mult, 23
print.Trajectory.NL, 24
print.Trajectory.POIS, 24
print.Trajectory.ZIP, 25
propAssign, 26

trajeR, 26
trajeR.BETA, 30
trajeR.CNORM, 32
trajeR.LOGIT, 35
trajeR.NL, 37
trajeR.POIS, 40
trajeR.ZIP, 42
trajeR_mult, 45
trajeRAIC, 46
trajeRBIC, 47
trajeRSH, 47