Package 'trajmsm'

September 17, 2024

Type Package

Title Marginal Structural Models with Latent Class Growth Analysis of Treatment Trajectories

Version 0.1.2

Maintainer Awa Diop <awa.diop.2@ulaval.ca>

Description Implements marginal structural models combined with a latent class growth analysis framework for assessing the causal effect of treatment trajectories. Based on the approach described in ``Marginal Structural Models with Latent Class Growth Analysis of Treatment Trajectories" Diop, A., Sirois, C., Guertin, J.R., Schnitzer, M.E., Candas, B., Cossette, B., Poirier, P., Brophy, J., Mésidor, M., Blais, C. and Hamel, D., (2023) <doi:10.1177/09622802231202384>.

License GPL (>= 3)

Encoding UTF-8

Imports stats, e1071, flexmix, ggplot2, survival, sandwich, utils

RoxygenNote 7.3.1

URL https://github.com/awamaeva/R-package-trajmsm

BugReports https://github.com/awamaeva/R-package-trajmsm/issues

NeedsCompilation no

Author Awa Diop [aut, cre],
Denis Talbot [aut]

Repository CRAN

Date/Publication 2024-09-17 03:50:02 UTC

Contents

`build_traj` *Wrapper for flexmix*

Description

Call the package flexmix to build trajectory groups

Usage

```
build_traj(
  obsdata,
  formula,
  number_traj,
  identifier,
  family = "binomial",
  seed = 945,
  control = list(iter.max = 1000, minprior = 0),
  ...
)
```

Arguments

`obsdata`	Data to build trajectory groups in long format.
`formula`	Designate the formula to model the longitudinal variable of interest.
`number_traj`	An integer to fix the number of trajectory groups.
`identifier`	A string to designate the column name for the unique identifier.
`family`	Designate the type of distribution ("gaussian", "binomial", "poisson", "gamma").
`seed`	Set a seed for replicability.
`control`	Object of class FLXcontrol.
`...`	Additional arguments passed to the flexmix function.

Value

A list containing the posterior probability matrix and the fitted trajectory model.

Examples

```
obsdata_long = gendata(n = 1000,format = "long", total_followup = 6, seed = 945)
formula = as.formula(cbind(statins, 1 - statins) ~ time)
restraj = build_traj(obsdata = obsdata_long, number_traj = 3, formula = formula, identifier = "id")
```

`gendata` *Generate data trajectories for MSM*

Description

Provides datasets for running examples for LCGA-MSM and LCGA-HRMSM.

Usage

```
gendata(
  n,
  include_censor = FALSE,
  format = c("long", "wide"),
  start_year = 2011,
  total_followup,
  timedep_outcome = FALSE,
  seed
)
```

Arguments

`n`	Number of observations to generate.
`include_censor`	Logical, if TRUE, includes censoring.
`format`	Character, either "long" or "wide" for the format of the output data frame.
`start_year`	Baseline year.
`total_followup`	Number of measuring times.
`timedep_outcome`	Logical, if TRUE, includes a time-dependent outcome.
`seed`	Use a specific seed value to ensure the simulated data is replicable.

Value

A data frame with generated data trajectories.

Examples

```
gendata(n = 100, include_censor = FALSE, format = "wide",total_followup = 3, seed = 945)
```

gformula *Counterfactual means via G-Formula*

Description

Calculates counterfactual means using the g-formula approach.

Usage

```
gformula(
  formula,
  baseline,
  covariates,
  treatment,
  outcome,
  ntimes_interval,
  obsdata
)
```

Arguments

formula	Specification of the model for the outcome to be fitted.
baseline	Names of the baseline covariates.
covariates	Names of the time-varying covariates (should be a list).
treatment	Names of the time-varying treatment.
outcome	Name of the outcome variable.
ntimes_interval	Length of a time-interval (s).
obsdata	Observed data in wide format.

Value

list_gform_countermeans
: List of counterfactual means obtained with g-formula.

Author(s)

Awa Diop, Denis Talbot

Examples

```
obsdata = gendata(n = 1000, format = "wide", total_followup = 6, seed = 945)
years <- 2011:2016
baseline_var <- c("age","sex")
variables <- c("hyper", "bmi")
var_cov <- c("statins","hyper", "bmi")
covariates <- lapply(years, function(year) {
```

```
paste0(variables, year)})
treatment_var <- paste0("statins", 2011:2016)
formula = paste0("y ~", paste0(treatment_var,collapse = "+"), "+",
                paste0(unlist(covariates), collapse = "+"),"+",
                paste0(baseline_var, collapse = "+"))
res_gform <- gformula(formula = formula, baseline = baseline_var, covariates = covariates,
treatment = treatment_var, outcome = "y", ntimes_interval = 6, obsdata =   obsdata )
```

---

ggtraj *ggplot Trajectory*

Description

Use "ggplot2" to plot trajectory groups produced by the function "build_traj" using the observed treatment.

Usage

```
ggtraj(traj_data, treatment, time, identifier, class, FUN = mean, ...)
```

Arguments

traj_data	Merged datasets containing observed data in long format and trajectory groups.
treatment	Name of the time-varying treatment.
time	Name of the time variable.
identifier	Name of the identifier variable.
class	Name of the trajectory groups.
FUN	Specify which statistics to display, by default calculate the mean.
...	Additional arguments to be passed to ggplot functions.

Value

A ggplot object representing the trajectory groups using the observed treatment.

Examples

```
obsdata_long = gendata(n = 1000, format = "long", total_followup = 12, seed = 945)
restraj = build_traj(obsdata = obsdata_long, number_traj = 3,
formula = as.formula(cbind(statins, 1 - statins) ~ time), identifier = "id")
datapost = restraj$data_post
head(datapost)
traj_data_long <- merge(obsdata_long, datapost, by = "id")
    AggFormula <- as.formula(paste("statins", "~", "time", "+", "class"))
    Aggtraj_data <- aggregate(AggFormula, data = traj_data_long, FUN = mean)
    Aggtraj_data
#Aggtraj_data with labels
```

```
traj_data_long[ , "traj_group"] <- factor(ifelse(traj_data_long[ , "class"] == "3" ,"Group1" ,
ifelse (traj_data_long[ , "class"]== "1" , "Group2" ,"Group3")))
AggFormula <- as.formula(paste("statins", "~", "time", "+", "traj_group"))
Aggtraj_data <- aggregate(AggFormula, data = traj_data_long, FUN = mean)
ggtraj(traj_data =  Aggtraj_data,
treatment = "statins",time= "time",identifier="id",class = "traj_group", FUN = mean)
```

`inverse_probability_weighting` *Inverse Probability Weighting*

Description

Compute stabilized and unstabilized weights, with or without censoring.

Usage

```
inverse_probability_weighting(
  numerator = c("stabilized", "unstabilized"),
  identifier,
  baseline,
  covariates,
  treatment,
  include_censor = FALSE,
  censor,
  obsdata
)
```

Arguments

`numerator`	To choose between stabilized and unstabilized weights.
`identifier`	Name of the column of the unique identifier.
`baseline`	Name of the baseline covariates.
`covariates`	Name of the time-varying covariates.
`treatment`	Name of the time-varying treatment.
`include_censor`	Logical value TRUE/FALSE to include or not a censoring variable.
`censor`	Name of the censoring variable.
`obsdata`	Observed data in wide format.

Value

Inverse Probability Weights (Stabilized and Unstabilized) with and without censoring.

Author(s)

Awa Diop, Denis Talbot

Examples

```
obsdata = gendata(n = 1000, format = "wide",total_followup = 3, seed = 945)
baseline_var <- c("age","sex")
covariates <- list(c("hyper2011", "bmi2011"),
c("hyper2012", "bmi2012"),c("hyper2013", "bmi2013"))
treatment_var <- c("statins2011","statins2012","statins2013")
stabilized_weights = inverse_probability_weighting(numerator = "stabilized",
identifier = "id", covariates = covariates, treatment = treatment_var,
baseline = baseline_var, obsdata = obsdata)
```

---

pltmle *Counterfactual means for a Pooled LTMLE*

Description

Function to estimate counterfactual means for a pooled LTMLE.

Usage

```
pltmle(
  formula,
  outcome,
  treatment,
  covariates,
  baseline,
  ntimes_interval,
  number_traj,
  time,
  time_values,
  identifier,
  obsdata,
  traj,
  total_followup,
  treshold = treshold,
  class_var,
  class_pred
)
```

Arguments

formula	Specification of the model for the outcome to be fitted.
outcome	Name of the outcome variable.
treatment	Time-varying treatment.
covariates	Covariates.
baseline	Name of baseline covariates.

`ntimes_interval`
: Length of a time-interval (s).

`number_traj`
: An integer to choose the number of trajectory groups.

`time`
: Name of the time variable.

`time_values`
: Measuring times.

`identifier`
: Name of the column of the unique identifier.

`obsdata`
: Observed data in wide format.

`traj`
: Matrix of indicators for the trajectory groups.

`total_followup`
: Number of measuring times per interval.

`treshold`
: For weight truncation.

`class_var`
: Name of the trajectory group variable.

`class_pred`
: Vector of predicted trajectory groups.

Value

`list_pltmle_countermeans`
: Counterfactual means and influence functions with the pooled ltmle.

`D`
: Influence functions

Author(s)

Awa Diop, Denis Talbot

Examples

```
obsdata_long = gendata(n = 2000, format = "long",total_followup = 3, seed = 945)
baseline_var <- c("age","sex")
covariates <- list(c("hyper2011", "bmi2011"),
c("hyper2012", "bmi2012"),c("hyper2013", "bmi2013"))
treatment_var <- c("statins2011","statins2012","statins2013")
time_values <- c(2011,2012,2013)
formulaA = as.formula(cbind(statins, 1 - statins) ~ time)
restraj = build_traj(obsdata = obsdata_long, number_traj = 3,
formula = formulaA, identifier = "id")
datapost = restraj$data_post
trajmsm_long <- merge(obsdata_long, datapost, by = "id")
    AggFormula <- as.formula(paste("statins", "~", "time", "+", "class"))
    AggTrajData <- aggregate(AggFormula, data = trajmsm_long, FUN = mean)
    AggTrajData
trajmsm_long[ , "traj_group"] <- trajmsm_long[ , "class"]
obsdata= reshape(trajmsm_long, direction = "wide", idvar = "id",
v.names = c("statins","bmi","hyper"), timevar = "time", sep ="")
formula =  as.formula(" y ~ statins2011 + statins2012 + statins2013 +
hyper2011 + bmi2011 + hyper2012 + bmi2012 +
 hyper2013 + bmi2013 + age + sex ")
class = factor(predict_traj(identifier = "id", total_followup = 3,
        treatment = "statins", time = "time", time_values = time_values,
        trajmodel = restraj$traj_model)$post_class);
```

```
traj=t(sapply(1:8,function(x)sapply(1:3,function(i)ifelse(class[x]==i,1,0))))
traj[,1]=1
res_pltmle = pltmle(formula = formula, outcome = "y",treatment = treatment_var,
covariates = covariates, baseline = baseline_var, ntimes_interval = 3, number_traj = 3,
 time =  "time",time_values = time_values,identifier = "id",obsdata = obsdata,
traj=traj, treshold = 0.99, class_pred= class, class_var = "class")
res_pltmle$counter_means
```

`predict_traj` *Predict trajectory groups for deterministic treatment regimes*

Description

Function to predict trajectory groups for deterministic treatment regimes used with gformula and pooled LTMLE.

Usage

```
predict_traj(
  identifier,
  total_followup,
  treatment,
  time,
  time_values,
  trajmodel
)
```

Arguments

`identifier`	Name of the column of the unique identifier.
`total_followup`	Number of measuring times.
`treatment`	Name of the time-varying treatment.
`time`	Name of the variable time.
`time_values`	Values of the time variable.
`trajmodel`	Trajectory model built with the observed treatment.

Value

A data.frame with the posterior probabilities.

Author(s)

Awa Diop, Denis Talbot

split_data *Split observed data into multiple subsets*

Description

Function to split the data into multiple subsets of size s each one subset corresponding to one time-interval.

Usage

```
split_data(
  obsdata,
  total_followup,
  ntimes_interval,
  time,
  time_values,
  identifier
)
```

Arguments

obsdata	Observed data in wide format.
total_followup	Total length of follow-up.
ntimes_interval	Number of measuring times per interval.
time	Name of the time variable.
time_values	Measuring times.
identifier	Identifier of individuals.

Value

all_df	All subsets, list of time intervals.

Author(s)

Awa Diop Denis Talbot

Examples

```
obsdata = gendata(n = 1000, format = "long", total_followup = 8, seed = 945)
years <- 2011:2018
res = split_data(obsdata = obsdata, total_followup = 8,
ntimes_interval = 6,time = "time", time_values = years,identifier = "id")
```

`trajhrmsm_gform` *History Restricted MSM and Latent Class of Growth Analysis estimated with G-formula.*

Description

Estimate parameters of LCGA-HRMSM using g-formula. and bootstrap to get standard errors.

Usage

```
trajhrmsm_gform(
  degree_traj = c("linear", "quadratic", "cubic"),
  rep = 50,
  treatment,
  covariates,
  baseline,
  outcome,
  ntimes_interval,
  total_followup,
  time,
  time_values,
  identifier,
  var_cov,
  number_traj = 3,
  family = "poisson",
  obsdata
)
```

Arguments

`degree_traj`	To specify the polynomial degree for modelling the time-varying treatment.
`rep`	Number of repetition for the bootstrap.
`treatment`	Name of the time-varying treatment.
`covariates`	Names of the time-varying covariates (should be a list).
`baseline`	Name of baseline covariates.
`outcome`	Name of the outcome variable.
`ntimes_interval`	Length of a time-interval (s).
`total_followup`	Total length of follow-up.
`time`	Name of the time variable.
`time_values`	Measuring times.
`identifier`	Name of the column of the unique identifier.
`var_cov`	Names of the time-varying covariates.
`number_traj`	Number of trajectory groups.

family	Specification of the error distribution and link function to be used in the model.
obsdata	Data in a long format.

Value

A list containing the following components:

results_hrmsm_gform Matrix of estimates for LCGA-MSM, obtained using the g-formula method.

result_coef_boot Matrix of estimates obtained with bootstrap.

restraj Fitted trajectory model.

mean_adh Matrix of mean adherence per trajectory group.

Author(s)

Awa Diop Denis Talbot

Examples

```
obsdata_long = gendata(n = 5000, format = "long", total_followup = 8,
timedep_outcome = TRUE,  seed = 845)
baseline_var <- c("age","sex")
years <- 2011:2018
variables <- c("hyper", "bmi")
covariates <- lapply(years, function(year) {
paste0(variables, year)})
treatment_var <- paste0("statins", 2011:2018)
var_cov <- c("statins","hyper", "bmi")
reshrmsm_gform = trajhrmsm_gform(degree_traj = "linear", rep=50 ,
treatment = treatment_var,covariates = covariates, baseline = baseline_var,
outcome = "y",var_cov = var_cov, ntimes_interval = 6, total_followup = 8,
 time = "time",time_values = years, identifier = "id",
number_traj = 3, family = "poisson", obsdata = obsdata_long)
reshrmsm_gform$results_hrmsm_gform
```

---

trajhrmsm_ipw *History Restricted MSM and Latent Class of Growth Analysis estimated with IPW.*

Description

Estimate parameters of LCGA-HRMSM using IPW.

Usage

```
trajhrmsm_ipw(
  degree_traj = c("linear", "quadratic", "cubic"),
  numerator = c("stabilized", "unstabilized"),
  identifier,
  baseline,
  covariates,
  treatment,
  outcome,
  var_cov,
  include_censor = FALSE,
  ntimes_interval,
  total_followup,
  time,
  time_values,
  family = "poisson",
  censor = censor,
  number_traj,
  obsdata,
  weights = NULL,
  treshold = 0.999
)
```

Arguments

degree_traj	To specify the polynomial degree for modelling the time-varying treatment.
numerator	To choose between stabilized and unstabilized weights.
identifier	Name of the column of the unique identifier.
baseline	Names of the baseline covariates.
covariates	Names of the time-varying covariates (should be a list).
treatment	Name of the time-varying treatment.
outcome	Name of the outcome variable.
var_cov	Names of the time-varying covariates.
include_censor	Logical, if TRUE, includes censoring.
ntimes_interval	Length of a time-interval (s).
total_followup	Total length of follow-up.
time	Name of the time variable.
time_values	Values of the time variable.
family	specification of the error distribution and link function to be used in the model.
censor	Name of the censoring variable.
number_traj	Number of trajectory groups.
obsdata	Data in a long format.

weights	A vector of estimated weights. If NULL, the weights are computed by the function.
treshold	For weight truncation.

Value

Provides a matrix of estimates for LCGA-HRMSM, obtained using IPW.

Author(s)

Awa Diop, Denis Talbot

Examples

```
obsdata_long = gendata(n = 5000, format = "long", total_followup = 8,
timedep_outcome = TRUE,  seed = 845)
baseline_var <- c("age","sex")
years <- 2011:2018
variables <- c("hyper", "bmi")
covariates <- lapply(years, function(year) {
paste0(variables, year)})
treatment_var <- paste0("statins", 2011:2018)
var_cov <- c("statins","hyper", "bmi","y")
reshrmsm_ipw <- trajhrmsm_ipw(degree_traj = "linear", numerator = "stabilized",
identifier = "id", baseline = baseline_var,
covariates = covariates, treatment = treatment_var,
outcome = "y", var_cov= var_cov,include_censor = FALSE,
 ntimes_interval = 6,total_followup = 8, time = "time", time_values = 2011:2018,
family = "poisson", number_traj = 3, obsdata = obsdata_long, treshold = 0.999)
reshrmsm_ipw$res_trajhrmsm_ipw
```

---

trajhrmsm_pltmle *History Restricted MSM and Latent Class of Growth Analysis estimated with a Pooled LTMLE.*

Description

Estimate parameters of LCGA-HRMSM using a Pooled LTMLE.

Usage

```
trajhrmsm_pltmle(
  degree_traj = c("linear", "quadratic", "cubic"),
  treatment,
  covariates,
  baseline,
  outcome,
  ntimes_interval,
```

```
  total_followup,
  time,
  time_values,
  identifier,
  var_cov,
  number_traj = 3,
  family = "poisson",
  obsdata,
  treshold = 0.99
)
```

Arguments

degree_traj	To specify the polynomial degree for modelling the time-varying treatment.
treatment	Name of time-varying treatment.
covariates	Names of time-varying covariates (should be a list).
baseline	Names of baseline covariates.
outcome	Name of the outcome variable.
ntimes_interval	Length of a time-interval (s).
total_followup	Total length of follow-up.
time	Name of the time variable.
time_values	Measuring times.
identifier	Name of the column for unique identifiant.
var_cov	Names of the time-varying covariates.
number_traj	Number of trajectory groups.
family	Specification of the error distribution and link function to be used in the model.
obsdata	Data in a long format.
treshold	For weight truncation.

Value

A list containing the following components:

results_hrmsm_pltmle Matrix of estimates for LCGA-HRMSM, obtained using the pooled ltlmle method.

restraj Fitted trajectory model.

mean_adh Matrix of the mean adherence per trajectory group.

Author(s)

Awa Diop Denis Talbot

Examples

```
obsdata_long = gendata(n = 5000, format = "long",
total_followup = 8, timedep_outcome = TRUE,  seed = 845)
baseline_var <- c("age","sex")
years <- 2011:2018
variables <- c("hyper", "bmi")
covariates <- lapply(years, function(year) {
  paste0(variables, year)})
treatment_var <- paste0("statins", 2011:2018)
var_cov <- c("statins","hyper", "bmi","y")
respltmle = trajhrmsm_pltmle(degree_traj = "linear", treatment = treatment_var,
covariates = covariates, baseline = baseline_var,
outcome = paste0("y", 2016:2018),var_cov = var_cov, ntimes_interval = 6,
total_followup = 8, time = "time",time_values = years, identifier = "id",
number_traj = 3, family = "poisson", obsdata = obsdata_long,treshold = 1)
respltmle$results_hrmsm_pltmle
```

`trajmsm_gform` *Parametric g-formula*

Description

Estimate parameters of LCGA-MSM using g-formula and bootstrap to get standard errors.

Usage

```
trajmsm_gform(
  formula = formula,
  rep = 50,
  identifier,
  baseline,
  covariates,
  treatment,
  outcome,
  total_followup,
  time = time,
  time_values,
  var_cov,
  trajmodel,
  ref,
  obsdata
)
```

Arguments

`formula`	Specification of the model for the outcome to be fitted.
`rep`	Number of repetitions for the bootstrap.

identifier	Name of the column of the unique identifier.
baseline	Vector of names of the baseline covariates.
covariates	List of names of the time-varying covariates.
treatment	Vector of names of the time-varying treatment.
outcome	Name of the outcome of interest.
total_followup	Total length of follow-up.
time	Name of the time variable.
time_values	Measuring times.
var_cov	Names of the time-varying covariates.
trajmodel	Trajectory model built with the observed treatment.
ref	The reference trajectory group.
obsdata	Observed data in wide format.

Value

Provides a matrix of estimates for LCGA-MSM, obtained using the g-formula method.

Author(s)

Awa Diop Denis Talbot

Examples

```
obsdata_long = gendata(n = 1000, format = "long", total_followup = 6, seed = 845)
years <- 2011:2016
baseline_var <- c("age","sex")
variables <- c("hyper", "bmi")
var_cov <- c("statins","hyper", "bmi")
covariates <- lapply(years, function(year) {
paste0(variables, year)})
treatment_var <- paste0("statins", 2011:2016)
formula_treatment = as.formula(cbind(statins, 1 - statins) ~ time)
restraj = build_traj(obsdata = obsdata_long, number_traj = 3,
formula = formula_treatment, identifier = "id")
datapost = restraj$data_post
trajmsm_long <- merge(obsdata_long, datapost, by = "id")
    AggFormula <- as.formula(paste("statins", "~", "time", "+", "class"))
    AggTrajData <- aggregate(AggFormula, data = trajmsm_long, FUN = mean)
    AggTrajData
obsdata = reshape(data = trajmsm_long, direction = "wide", idvar = "id",
v.names = c("statins","bmi","hyper"), timevar = "time", sep ="")
formula = paste0("y ~", paste0(treatment_var,collapse = "+"), "+",
                paste0(unlist(covariates), collapse = "+"),"+",
                paste0(baseline_var, collapse = "+"))
resmsm_gform <- trajmsm_gform(formula = formula, identifier = "id",rep = 5,
baseline = baseline_var, covariates = covariates, var_cov = var_cov,
treatment = treatment_var, outcome = "y", total_followup = 6,time = "time",
time_values = years, trajmodel = restraj$traj_model,ref = "1", obsdata =   obsdata )
```

resmsm_gform

---

trajmsm_ipw *Marginal Structural Model and Latent Class of Growth Analysis estimated with IPW*

Description

Estimate parameters of LCGA-MSM using IPW.

Usage

```
trajmsm_ipw(
  formula1,
  formula2,
  family,
  identifier,
  treatment,
  covariates,
  baseline,
  obsdata,
  numerator = "stabilized",
  include_censor = FALSE,
  censor,
  weights = NULL,
  treshold = 0.99
)
```

Arguments

formula1	Specification of the model for the outcome to be fitted for a binomial or gaussian distribution.
formula2	Specification of the model for the outcome to be fitted for a survival outcome.
family	Specification of the error distribution and link function to be used in the model.
identifier	Name of the column of the unique identifier.
treatment	Time-varying treatment.
covariates	Names of the time-varying covariates (should be a list).
baseline	Name of the baseline covariates.
obsdata	Dataset to be used in the analysis.
numerator	Type of weighting ("stabilized" or "unstabilized").
include_censor	Logical, if TRUE, includes censoring.
censor	Name of the censoring variable.
weights	A vector of estimated weights. If NULL, the weights are computed by the function IPW.
treshold	For weight truncation.

Value

Provides a matrix of estimates for LCGA-MSM, obtained using IPW.

Provides a matrix of estimates for LCGA-MSM, obtained using IPW.

Examples

```
obsdata_long = gendata(n = 1000, format = "long", total_followup = 6, seed = 845)
years <- 2011:2016
baseline_var <- c("age","sex")
variables <- c("hyper", "bmi")
covariates <- lapply(years, function(year) {
paste0(variables, year)})
treatment_var <- paste0("statins", 2011:2016)
formula_treatment = as.formula(cbind(statins, 1 - statins) ~ time)
restraj = build_traj(obsdata = obsdata_long, number_traj = 3,
formula = formula_treatment, identifier = "id")
datapost = restraj$data_post
trajmsm_long <- merge(obsdata_long, datapost, by = "id")
    AggFormula <- as.formula(paste("statins", "~", "time", "+", "class"))
    AggTrajData <- aggregate(AggFormula, data = trajmsm_long, FUN = mean)
    AggTrajData
trajmsm_long$ipw_group <- relevel(trajmsm_long$class, ref = "1")
obsdata = reshape(data = trajmsm_long, direction = "wide", idvar = "id",
v.names = c("statins","bmi","hyper"), timevar = "time", sep ="")
formula = paste0("y ~", paste0(treatment_var,collapse = "+"), "+",
                paste0(unlist(covariates), collapse = "+"),"+",
                paste0(baseline_var, collapse = "+"))

resmsm_ipw = trajmsm_ipw(formula1 = as.formula("y ~ ipw_group"),
           identifier = "id", baseline = baseline_var, covariates = covariates,
           treatment = treatment_var, family = "binomial",
       obsdata = obsdata,numerator = "stabilized", include_censor = FALSE, treshold = 0.99)
resmsm_ipw
```

`trajmsm_pltmle` *Pooled LTMLE*

Description

Estimate parameters of LCGA-MSM using pooled LTMLE with influence functions to estimate standard errors.

Usage

```
trajmsm_pltmle(
  formula = formula,
  identifier,
  baseline,
```

```
  covariates,
  treatment,
  outcome,
  number_traj,
  total_followup,
  time,
  time_values,
  trajmodel,
  ref,
  treshold = 0.99,
  obsdata,
  class_var
)
```

Arguments

formula	Specification of the model for the outcome to be fitted.
identifier	Name of the column for unique identifiant.
baseline	Names of the baseline covariates.
covariates	Names of the time-varying covariates (should be a list).
treatment	Name of the time-varying treatment.
outcome	Name of the outcome variable.
number_traj	An integer to choose the number of trajectory groups.
total_followup	Total length of follow-up.
time	Name of the time variable.
time_values	Measuring times.
trajmodel	Trajectory model built with the observed treatment.
ref	The reference group.
treshold	For weight truncation.
obsdata	Observed data in wide format.
class_var	Name of the trajectory group variable.

Value

Provides a matrix of estimates for LCGA-MSM, obtained using the pooled ltlmle method.

results_msm_pooledltmle	Estimates of a LCGA-MSM with pooled LTMLE.

Author(s)

Awa Diop, Denis Talbot

Examples

```
obsdata_long = gendata(n = 1000, format = "long", total_followup = 6, seed = 845)
years <- 2011:2016
baseline_var <- c("age","sex")
variables <- c("hyper", "bmi")
covariates <- lapply(years, function(year) {
paste0(variables, year)})
treatment_var <- paste0("statins", 2011:2016)
formula_treatment = as.formula(cbind(statins, 1 - statins) ~ time)
restraj = build_traj(obsdata = obsdata_long, number_traj = 3,
formula = formula_treatment, identifier = "id")
datapost = restraj$data_post
trajmsm_long <- merge(obsdata_long, datapost, by = "id")
    AggFormula <- as.formula(paste("statins", "~", "time", "+", "class"))
    AggTrajData <- aggregate(AggFormula, data = trajmsm_long, FUN = mean)
trajmsm_wide = reshape(data = trajmsm_long, direction = "wide", idvar = "id",
v.names = c("statins","bmi","hyper"), timevar = "time", sep ="")
formula = paste0("y ~", paste0(treatment_var,collapse = "+"), "+",
                paste0(unlist(covariates), collapse = "+"),"+",
                paste0(baseline_var, collapse = "+"))
resmsm_pltmle <- trajmsm_pltmle(formula = formula, identifier = "id",
 baseline = baseline_var,
 covariates = covariates, treatment = treatment_var,
 outcome = "y", time = "time", time_values = years,
 number_traj = 3, total_followup = 6,
 trajmodel = restraj$traj_model, ref = "1", obsdata = trajmsm_wide,
  treshold = 1,class_var = "class")
 resmsm_pltmle
```

Index