Package 'remstimate'

May 13, 2026

Type Package

Title Optimization Frameworks for Tie-Oriented and Actor-Oriented Relational Event Models

Version 3.0.0

Date 2026-05-12

Maintainer Giuseppe Arena <g.arena@uva.nl>

Description A comprehensive set of tools designed for optimizing likelihood within a tie-oriented (Butts, C., 2008, <doi:10.1111/j.1467-9531.2008.00203.x>) or an actor-oriented modelling framework (Stadtfeld, C., & Block, P., 2017, <doi:10.15195/v4.a14>) in relational event networks. The package accommodates both frequentist and Bayesian approaches. Maximum Likelihood Optimization (MLE) is supported. Bayesian estimation is done via Hamiltonian Monte Carlo (HMC).

License MIT + file LICENSE

URL https://tilburgnetworkgroup.github.io/remstimate/

BugReports https://github.com/TilburgNetworkGroup/remstimate/issues

Depends R (>= 4.0.0)

Imports Rcpp, remify (>= 4.0.0), remstats (>= 4.0.0), trust, mvnfast

Suggests knitr, rmarkdown, tinytest, survival

LinkingTo Rcpp, RcppArmadillo, remify (>= 4.0.0)

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

LazyDataCompression gzip

NeedsCompilation yes

Author Giuseppe Arena [aut, cre] (ORCID: <https://orcid.org/0000-0001-5204-3326>),
Rumana Lakdawala [aut],
Fabio Generoso Vieira [aut],

Joris Mulder [aut],
Marlyne Meijerink-Bosman [ctb],
Diana Karimova [ctb],
Mahdi Shafiee Kamalabad [ctb],
Roger Leenders [ctb]

Repository CRAN

Date/Publication 2026-05-13 05:10:20 UTC

Contents

`AIC.remstimate` *AIC for remstimate objects*

Description

Returns the AIC (Akaike's Information Criterion) value of a 'remstimate' object.

Usage

```
## S3 method for class 'remstimate'
AIC(object, ...)
```

Arguments

`object`	a `remstimate` object.
`...`	further arguments passed to `AIC`.

Value

AIC value.

AICC *AICC*

Description

A function that returns the AICC (Akaike's Information Corrected Criterion) value in a 'remstimate' object.

Usage

```
AICC(object, ...)

## S3 method for class 'remstimate'
AICC(object, ...)
```

Arguments

object	is a remstimate object.
...	further arguments to be passed to the 'AICC' method.

Value

AICC value of a 'remstimate' object.

Methods (by class)

- AICC(remstimate): AICC (Akaike's Information Corrected Criterion) value of a 'remstimate' object

Examples

```
# ----------------------------------- #
#       tie-oriented model: "MLE"     #
# ----------------------------------- #

# loading data
data(tie_data)

# processing event sequence with remify
tie_reh <- remify::remify(edgelist = tie_data$edgelist, model = "tie")

# specifying linear predictor
tie_model <- ~ 1 +
               remstats::indegreeSender()+
               remstats::inertia()+
               remstats::reciprocity()

# calculating statistics
tie_reh_stats <- remstats::remstats(reh = tie_reh,
```

```
                                    tie_effects = tie_model)

# running estimation
tie_mle <- remstimate::remstimate(reh = tie_reh,
                                  stats = tie_reh_stats,
                                  method = "MLE",
                                  ncores = 1)

# AICC
AICC(tie_mle)
```

---

ao_data *Actor-Oriented Relational Event History*

Description

A randomly generated sequence of relational events with 5 actors and 100 events. The event sequence is generated by following an actor-oriented modeling approach (for more information on the algorithm used for the generation, refer to `help(topic = remulateActor, package = "remulate")` or `?remulate::remulateActor`).

Usage

```
data(ao_data)
```

Format

`ao_data` is a list object containing the following objects:

`edgelist` a `data.frame` with the raw simulated edgelist. The columns of the `data.frame` are:

- `time` the timestamp indicating the time at which each event occurred
- `actor1` the actor that generated the relational event
- `actor2` the actor that received the relational event

`seed` the seed value used in `remulate::remulateActor()` for generating the event sequence

`true.pars` a list of two vectors named `"rate_model"` and `"choice_model"`, each containing the values of the parameters used in the generation of the event sequence

Examples

```
# (1) load the data into the workspace
data(ao_data)

# (2) process event sequence with \code{remify}
ao_reh <- remify::remify(edgelist = ao_data$edgelist, model = "actor")

# (3) define linear predictor and claculate stastistcs with \code{remstats} package
```

```
## linear predictor for the rate model
rate_model <- ~ 1 + remstats::indegreeSender()

## linear predictror for the choice model
choice_model <- ~ remstats::inertia() + remstats::reciprocity()

## calculate statistics
ao_reh_stats <- remstats::remstats(reh = ao_reh, sender_effects = rate_model,
receiver_effects = choice_model)

# (4) estimate model using method = "MLE" and print out summary

## estimate model
mle_ao <- remstimate::remstimate(reh = ao_reh, stats = ao_reh_stats, method = "MLE")

## print out a summary of the estimation
summary(mle_ao)
```

`BIC.remstimate` *BIC for remstimate objects*

Description

Returns the BIC (Bayesian Information Criterion) value of a 'remstimate' object.

Usage

```
## S3 method for class 'remstimate'
BIC(object, ...)
```

Arguments

`object`	a `remstimate` object.
`...`	further arguments passed to `BIC`.

Value

BIC value.

diagnostics *Compute the diagnostics of a* remstimate *object*

Description

Compute the diagnostics of a remstimate object

Usage

```
diagnostics(object, reh, stats, ...)

## S3 method for class 'remstimate'
diagnostics(object, reh, stats, top_pct = 0.05, ...)
```

Arguments

object	is a remstimate object.
reh	is a remify object, the same used for the 'remstimate' object.
stats	is a remstats object, the same used for the 'remstimate' object.
...	further arguments to be passed to the 'diagnostics' method.
top_pct	numeric scalar in (0,1): threshold for the top-percentage recall summary (default 0.05).

Methods (by class)

- diagnostics(remstimate): diagnostics of a 'remstimate' object

plot.diagnostics *Plot diagnostics of a* remstimate *object*

Description

Produces diagnostic plots from an object returned by diagnostics(). Plot 1 (waiting-time Q-Q) and plot 2 (Schoenfeld residuals) are computed from the diagnostics object alone. Plot 3 (recall scatter) is also derived from the diagnostics object. Plots 4 and 5 (posterior density and trace plots) are HMC-only and additionally require the original remstimate fit via object.

Usage

```
## S3 method for class 'diagnostics'
plot(
  x,
  object = NULL,
  which = c(1:3),
  effects = NULL,
  sender_effects = NULL,
  receiver_effects = NULL,
  n_per_page = 4L,
  ...
)
```

Arguments

x	a diagnostics object returned by diagnostics().
object	optional remstimate fit. Required for plots 4–5 (HMC posterior diagnostics); ignored otherwise.
which	integer vector of plots to produce. Default 1:3 covers waiting times, Schoenfeld residuals, and recall. Add 4 or 5 for HMC posterior density and trace plots.
effects	character vector of effect names to include (tie model). NULL uses all available effects.
sender_effects	character vector of sender-model effects (actor model). Pass NA to skip the sender model entirely.
receiver_effects	character vector of receiver-model effects (actor model). Pass NA to skip the receiver model entirely.
n_per_page	integer; maximum panels per page for multi-effect plots (Schoenfeld, posterior density, trace). Default 4L (2x2 grid).
...	further graphical arguments (currently unused).

Value

x invisibly.

---

plot.remstimate *Plot method for* remstimate *objects*

Description

Backward-compatible wrapper that computes diagnostics() when needed and delegates all plotting to plot.diagnostics. Existing call signatures continue to work unchanged.

Usage

```
## S3 method for class 'remstimate'
plot(
  x,
  reh,
  diagnostics = NULL,
  which = c(1:5),
  effects = NULL,
  sender_effects = NULL,
  receiver_effects = NULL,
  ...
)
```

Arguments

x	a remstimate object.
reh	a remify object used for the estimation.
diagnostics	optional pre-computed diagnostics object of class c("diagnostics","remstimate"). If NULL, stats must be supplied via ....
which	integer vector selecting plots (default 1:4).
effects	character vector of effects to plot (tie model).
sender_effects	character vector of sender-model effects (actor model).
receiver_effects	character vector of receiver-model effects (actor model).
...	pass stats here when diagnostics = NULL.

Value

x invisibly.

---

print.remstimate *Print out a quick overview of a* remstimate *object*

Description

A function that prints out the estimates returned by a 'remstimate' object.

Usage

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

Arguments

x	is a remstimate object.
...	further arguments to be passed to the print method.

Value

no return value. Prints out the main characteristics of a 'remstimate' object.

Examples

```
# ------------------------------------ #
#       method 'print' for the         #
# ------------------------------------ #

# loading data
data(ao_data)

# processing event sequence with remify
ao_reh <- remify::remify(edgelist = ao_data$edgelist, model = "actor")

# specifying linear predictor (for sender rate and receiver choice model)
rate_model <- ~ 1 + remstats::indegreeSender()
choice_model <- ~ remstats::inertia() + remstats::reciprocity()

# calculating statistics
ao_reh_stats <- remstats::remstats(reh = ao_reh,
                                   sender_effects = rate_model,
                                   receiver_effects = choice_model)

# running estimation
ao_mle <- remstimate::remstimate(reh = ao_reh,
                                 stats = ao_reh_stats,
                                 method = "MLE",
                                 nsim = 100,
                                 ncores = 1)

# print
ao_mle

# ------------------------------------ #
#   for more examples check vignettes  #
# ------------------------------------ #
```

`remstimate` *remstimate - MLE and HMC estimation for tie-oriented REM*

Description

Estimates tie-oriented relational event model parameters using Maximum Likelihood Estimation (`"MLE"`) or Hamiltonian Monte Carlo (`"HMC"`). Supports both full statistics (`tomstats`) and case-control sampled statistics (`tomstats_sampled`).

Usage

```
remstimate(
  reh,
  stats,
  method = c("MLE", "HMC"),
  ncores = 1L,
  nsim = 500L,
  nchains = 1L,
  burnin = 500L,
  thin = 10L,
  init = NULL,
  L = 50L,
  epsilon = 0.002,
  prior = NULL,
  seed = NULL,
  WAIC = FALSE,
  nsimWAIC = 100L,
  ...
)
```

Arguments

reh	a remify object.
stats	a tomstats or tomstats_sampled object (output of remstats::tomstats() or remstats::tomstats(sampling=TRUE)).
method	optimization method: "MLE" (default) or "HMC".
ncores	number of threads for parallelization (default 1).
nsim	[*HMC only*] number of MCMC iterations per chain after burnin. Default 500.
nchains	[*HMC only*] number of chains. Default 1.
burnin	[*HMC only*] burn-in iterations. Default 500.
thin	[*HMC only*] thinning interval. Default 10.
init	[*HMC only*] vector of initial parameter values. If NULL, MLE estimates are used as starting values.
L	[*HMC only*] number of leapfrog steps. Default 50.
epsilon	[*HMC only*] leapfrog step size. Default 0.002.
prior	[*HMC only*] list with elements mean (prior mean vector) and vcov (prior covariance matrix). If NULL, a standard normal prior is used.
seed	[*HMC only*] random seed for reproducibility.
WAIC	logical. Compute WAIC? Default FALSE. Only supported for MLE for now.
nsimWAIC	number of draws from the (approximate) posterior distribution used for the computation of the WAIC. Ignored if WAIC = FALSE. Default is 100L.
...	further arguments (currently unused).

Value

a remstimate S3 object.

Examples

```
# ------------------------------------ #
#       tie-oriented model: "MLE"      #
# ------------------------------------ #

# loading data
data(tie_data)

# processing event sequence with remify
tie_reh <- remify::remify(edgelist = tie_data$edgelist, model = "tie")

# specifying linear predictor
tie_model <- ~ 1 +
               remstats::indegreeSender()+
               remstats::inertia()+
               remstats::reciprocity()

# calculating statistics
tie_reh_stats <- remstats::remstats(reh = tie_reh,
                                    tie_effects = tie_model)

# running estimation
tie_mle <- remstimate::remstimate(reh = tie_reh,
                                  stats = tie_reh_stats,
                                  method = "MLE",
                                  ncores = 1)
# summary
summary(tie_mle)

# ------------------------------------ #
#      actor-oriented model: "MLE"     #
# ------------------------------------ #

# loading data
data(ao_data)

# processing event sequence with remify
ao_reh <- remify::remify(edgelist = ao_data$edgelist, model = "actor")

# specifying linear predictor (for sender rate and receiver choice model)
rate_model <- ~ 1 + remstats::indegreeSender()
choice_model <- ~ remstats::inertia() + remstats::reciprocity()

# calculating statistics
ao_reh_stats <- remstats::remstats(reh = ao_reh,
                                   sender_effects = rate_model,
                                   receiver_effects = choice_model)
```

```
# running estimation
ao_mle <- remstimate::remstimate(reh = ao_reh,
                                 stats = ao_reh_stats,
                                 method = "MLE",
                                 ncores = 1)
# summary
summary(ao_mle)

# ----------------------------------- #
#   for more examples check vignettes  #
# ----------------------------------- #
```

---

summary.remstimate *Generate the summary of a* remstimate *object*

Description

A function that returns the summary of a remstimate object.

Usage

```
## S3 method for class 'remstimate'
summary(object, ...)
```

Arguments

object	is a remstimate object.
...	further arguments to be passed to the 'summary' method.

Value

no return value. Prints out the summary of a 'remstimate' object. The output can be save in a list, which contains the information printed out by the summary method.

Examples

```
# ----------------------------------- #
#       method 'summary' for the       #
#         tie-oriented model.         #
# ----------------------------------- #

# loading data
data(tie_data)

# processing event sequence with remify
tie_reh <- remify::remify(edgelist = tie_data$edgelist, model = "tie")

# specifying linear predictor
```

```
tie_model <- ~ 1 +
               remstats::indegreeSender()+
               remstats::inertia()+
               remstats::reciprocity()

# calculating statistics
tie_reh_stats <- remstats::remstats(reh = tie_reh,
                                    tie_effects = tie_model)

# running estimation
tie_mle <- remstimate::remstimate(reh = tie_reh,
                                  stats = tie_reh_stats,
                                  method = "MLE",
                                  nsim = 100,
                                  ncores = 1)

# summary
summary(tie_mle)

# ------------------------------------ #
#      method 'summary' for the        #
#      actor-oriented model: "MLE"     #
# ------------------------------------ #

# loading data
data(ao_data)

# processing event sequence with remify
ao_reh <- remify::remify(edgelist = ao_data$edgelist, model = "actor")

# specifying linear predictor (for sender rate and receiver choice model)
rate_model <- ~ 1 + remstats::indegreeSender()
choice_model <- ~ remstats::inertia() + remstats::reciprocity()

# calculating statistics
ao_reh_stats <- remstats::remstats(reh = ao_reh,
                                   sender_effects = rate_model,
                                   receiver_effects = choice_model)

# running estimation
ao_mle <- remstimate::remstimate(reh = ao_reh,
                                 stats = ao_reh_stats,
                                 method = "MLE",
                                 nsim = 100,
                                 ncores = 1)

# summary
summary(ao_mle)

# ------------------------------------ #
#   for more examples check vignettes  #
# ------------------------------------ #
```

tie_data *Tie-Oriented Relational Event History*

Description

A randomly generated sequence of relational events with 5 actors and 100 events. The event sequence is generated by following a tie-oriented modeling approach (for more information run on console `help(topic = remulateTie, package = "remulate")` or `?remulate::remulateTie`).

Usage

```
data(tie_data)
```

Format

`tie_data` is a list object containing the following objects:

`edgelist` a `data.frame` with the raw simulated edgelist. The columns of the `data.frame` are:

- `time` the timestamp indicating the time at which each event occurred
- `actor1` the actor that generated the relational event
- `actor2` the actor that received the relational event

`seed` the seed value used in `remulate::remulateTie()` for generating the event sequence

`true.pars` a vector containing the values of the parameters used in the generation of the event sequence

Examples

```
# (1) load the data into the workspace
data(tie_data)

# (2) process event sequence with \code{remify}
tie_reh <- remify::remify(edgelist = tie_data$edgelist, model = "tie")

# (3) define linear predictor and claculate stastistcs with \code{remstats} package

## linear predictor
tie_model <- ~ 1 + remstats::indegreeSender() + remstats::inertia() + remstats::reciprocity()

## calculate statistics
tie_reh_stats <- remstats::remstats(reh = tie_reh, tie_effects = tie_model)

# (4) estimate model using method = "MLE" and print out summary

## estimate model
mle_tie <- remstimate::remstimate(reh = tie_reh, stats = tie_reh_stats, method = "MLE")

## print out a summary of the estimation
summary(mle_tie)
```

WAIC *WAIC*

Description

A function that returns the WAIC (Watanabe-Akaike's Information Criterion) value in a 'remstimate' object.

Usage

```
WAIC(object, ...)

## S3 method for class 'remstimate'
WAIC(object, ...)
```

Arguments

`object`	is a `remstimate` object.
`...`	further arguments to be passed to the 'WAIC' method.

Value

WAIC value of a 'remstimate' object.

Methods (by class)

- `WAIC(remstimate)`: WAIC (Watanabe-Akaike's Information Criterion) value of a 'remstimate' object

Examples

```
# No examples available at the moment
```

Index