Package 'Desimml'

March 19, 2026

Title Distributed Estimation in Single-Index Models with Multiple Links ('Desimml')

Version 0.1.4

Description Provides a user-friendly wrapper for fitting Distributed Single-Index Models with Multiple Links ('SIMML'). It estimates a linear combination (single-index) of baseline covariates X and models the treatment-specific outcome y via treatment-specific, non-parametrically estimated link functions. This allows flexible modelling of complex relationships between covariates, treatments and outcomes.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 3.5.0)

Imports stats, mgcv

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Guangbao Guo [aut, cre]

Maintainer Guangbao Guo <ggb11111111@163.com>

VignetteBuilder knitr

Repository CRAN

Date/Publication 2026-03-19 14:40:07 UTC

Contents

der.link *A subfunction used in estimation*

Description

This function computes the 1st derivative of the treatment-specific link function with respect to the single index, using finite difference.

Usage

```
der.link(g.fit, eps = 10^(-6))
```

Arguments

g.fit	a mgcv::gam object
eps	a small finite difference used in numerical differentiation.

Value

A numeric vector representing the 1st derivative of the link function evaluated at the single index.

See Also

fit.simml, simml

Examples

```
dat <- generate.data(n = 200, p = 5)

res <- simml(y = dat$y, A = dat$A, X = dat$X)

derivative <- der.link(res$g.fit)
head(derivative)
```

fit.simml *Workhorse function for Single-index models with multiple-links (SIMML)*

Description

Estimates a linear combination (a single-index) of covariates X, and models the treatment-specific outcome y via treatment-specific nonparametrically-defined link functions.

Usage

```
fit.simml(
  y,
  A,
  X,
  Xm = NULL,
  aug = NULL,
  rho = 0,
  family = "gaussian",
  R = NULL,
  bs = "ps",
  k = 8,
  sp = NULL,
  linear.link = FALSE,
  method = "GCV.Cp",
  gamma = 1,
  max.iter = 20,
  eps.iter = 0.01,
  trace.iter = TRUE
)
```

Arguments

y	a n-by-1 vector of treatment outcomes
A	a n-by-1 vector of treatment variable; each element is assumed to take a value on a continuum
X	a n-by-p matrix of baseline covariates
Xm	a n-by-q design matrix associated with an X main effect model; default is NULL
aug	augmentation matrix for regularization; default is NULL
rho	regularization parameter; default is 0
family	specifies the distribution of y; e.g., "gaussian", "binomial", "poisson"
R	the number of response categories for the case of family = "ordinal"; default is NULL
bs	basis type for the treatment (A) and single-index domains; default is "ps" (p-splines)
k	basis dimension for the treatment (A) and single-index domains; default is 8
sp	smoothing parameter for the treatment-specific link functions; if NULL, then estimated from the data
linear.link	if TRUE, the link function is restricted to be linear; default is FALSE
method	the smoothing parameter estimation method; default is "GCV.Cp"
gamma	multiplicative constant for smoothing parameter; default is 1
max.iter	maximum number of iterations; default is 20
eps.iter	convergence tolerance; default is 0.01
trace.iter	logical; if TRUE, iteration progress is printed; default is TRUE

Value

a list of information of the fitted SIMML

Examples

```
dat <- generate.data(n = 200, p = 5)

fit_obj <- fit.simml(y = dat$y, A = dat$A, X = dat$X, max.iter = 10)

print(fit_obj$beta.coef)
```

`generate.data` *generate.data generates an example dataset from a mean model that has a "main" effect component and a treatment-by-covariates interaction effect component (and a random component for noise).*

Description

generate.data generates an example dataset from a mean model that has a "main" effect component and a treatment-by-covariates interaction effect component (and a random component for noise).

Usage

```
generate.data(
  n = 200,
  p = 10,
  family = "gaussian",
  sigma = 0.4,
  sigmaX = 1,
  correlationX = 0,
  pi.1 = 0.5,
  s = 2,
  delta = 1
)
```

Arguments

`n`	sample size.
`p`	dimension of covariates.
`family`	specifies the distribution of the outcome y; "gaussian", "binomial", "poisson"; the defult is "gaussian"
`sigma`	standard deviation of the random noise term (for gaussian response).
`sigmaX`	standard deviation of covariates.
`correlationX`	correlation among covariates.

pi.1	probability for treatment assignment.
s	type of interaction effect.
delta	strength of main effect.

Value

y	a n-by-1 vector of treatment outcomes.
A	a n-by-1 vector of treatment indicators.
X	a n-by-p matrix of pretreatment covariates.
SNR	the "signal" (interaction effect) to "nuisance" (main effect) variance ratio (SNR) in the canonical parameter function.
true.beta	the true single-index coefficient vector.
true.eta	the true main effect coefficient vector.
optTr	a n-by-1 vector of treatments, indicating the optimal treatment selections.
value.opt	the "value" implied by the optimal treatment decision rule, optTr.

Examples

```
data <- generate.data(n = 100, p = 5)
head(data$y)
head(data$X)
```

---

ordinal.data *A function for ordinal categorical response data generation.*

Description

ordinal.data generates ordered category response data (with p covariates and a treatment variable).

Usage

```
ordinal.data(n = 400, p = 10, R = 11, delta = 1, s = "nonlinear", sigma = 0)
```

Arguments

n	sample size.
p	dimension of covariates.
R	number of response levels in y
delta	magnitude of "main" effect (i.e., "nuisance" effect) of the covariates; a large delta means a larger "nuisance" variance.
s	type of the treatment-by-covariates interation effect ("linear" or "nonlinear")
sigma	noise sd in the latent variable representation

Value

y	a n-by-1 vector of treatment outcomes.
A	a n-by-1 vector of treatment indicators.
X	a n-by-p matrix of pretreatment covariates.
SNR	the "signal" (interaction effect) to "nuisance" (main effect) variance ratio (SNR) in the canonical parameter function.
true.beta	the true single-index coefficient vector.
delta	magnitude of "main" effect.
s	type of the treatment-by-covariates interation effect.

Examples

```
ord_dat <- ordinal.data(n = 200, p = 5, R = 11)

table(ord_dat$y)

print(ord_dat$SNR)
```

---

pred.simml *SIMML prediction function*

Description

This function makes predictions from an estimated SIMML, given a (new) set of pretreatment covariates. The function returns a set of predicted outcomes for each treatment condition and a set of recommended treatment assignments (assuming a larger value of the outcome is better).

Usage

```
pred.simml(
  simml.obj,
  newX = NULL,
  newA = NULL,
  newXm = NULL,
  single.index = NULL,
  type = "link",
  maximize = TRUE
)
```

Arguments

simml.obj	a simml object
newX	a (n-by-p) matrix of new values for the covariates X at which predictions are to be made.

newA	a (n-by-L) matrix of new values for the treatment A at which predictions are to be made.
newXm	a (n-by-q) matrix of new values for the covariates associated with the fitted main effect Xm at which predictions are to be made.
single.index	a length n vector specifying new values for the single-index at which predictions are to be made; the default is NULL.
type	the type of prediction required; the default "response" is on the scale of the response variable; the alternative "link" is on the scale of the linear predictors.
maximize	the default is TRUE, assuming a larger value of the outcome is better; if FALSE, a smaller value is assumed to be prefered.

Value

pred.new	a (n-by-L) matrix of predicted values; each column represents a treatment option.
trt.rule	a (n-by-1) vector of suggested treatment assignments

Author(s)

Park, Petkova, Tarpey, Ogden

See Also

simml,fit.simml

Examples

```
train_dat <- generate.data(n = 200, p = 5)
fit <- simml(y = train_dat$y, A = train_dat$A, X = train_dat$X)

test_X <- matrix(rnorm(50 * 5), nrow = 50)
predictions <- pred.simml(simml.obj = fit, newX = test_X)

head(predictions$trt.rule)
```

---

simml *Single-index models with multiple-links (main function)*

Description

simml is the wrapper function for Single-index models with multiple-links (SIMML). The function estimates a linear combination (a single-index) of covariates X, and models the treatment-specific outcome y, via treatment-specific nonparametrically-defined link functions. This approach allows for flexible modeling of the complex relationships between covariates, treatments, and outcomes.

Usage

```
simml(y, A, X, family = "gaussian", bs = "cr", k = 3)
```

Arguments

y	a n-by-1 vector of treatment outcomes;
A	a n-by-1 vector of treatment variable; each element is assumed to take a value on a continuum.
X	a n-by-p matrix of baseline covarates.
family	specifies the distribution of y; e.g., "gaussian", "binomial", "poisson";
bs	basis type for the treatment (A) and single-index domains, respectively; the default is "ps" (p-splines);
k	basis dimension for the treatment (A) and single-index domains, respectively.

Value

A list containing the estimated single-index and the fitted GAM object.

Examples

```
dat <- generate.data(n = 200, p = 5, family = "gaussian")

res <- simml(y = dat$y, A = dat$A, X = dat$X, family = "gaussian")

print(res$beta.coef)
```

Index