Package ‘tramvs’

September 4, 2024

Type Package

Title Optimal Subset Selection for Transformation Models

Version 0.0-6

Description Greedy optimal subset selection for transformation models (Hothorn et al., 2018, <doi:10.1111/sjos.12291>) based on the abess algorithm (Zhu et al., 2020, <doi:10.1073/pnas.2014241117>). Applicable to models from packages 'tram' and 'cotram'.

Depends R (>= 4.0), tram (>= 0.6-1)

Imports stats, variables, methods, cotram, future, future.apply, mvtnorm

Suggests abess, tramnet, colorspace, knitr, mlt, TH.data, survival, ordinal, testthat (>= 3.0.0)

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.1

URL `http://ctm.R-forge.R-project.org`

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Lucas Kook [aut, cre],
Sandra Siegfried [ctb],
Torsten Hothorn [ctb]

Maintainer Lucas Kook <`lucasheinrich.kook@gmail.com`>

Repository CRAN

Date/Publication 2024-09-04 13:50:02 UTC

Contents

`abess_mmlt` *Optimal subset selection for multivariate transformation models*

Description

Optimal subset selection for multivariate transformation models

Usage

```
abess_mmlt(
  mltargs,
  supp,
  k_max = supp,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  m0 = NULL,
  ...
)
```

Arguments

mltargs	Arguments passed to mmlt
supp	support size of the coefficient vector
k_max	maximum support size to consider during the splicing algorithm. Defaults to supp.
thresh	threshold when to stop splicing. Defaults to 0.01 * supp * p * log(log(n)) / n$, where p denotes the number of predictors and n the sample size.
init	initialize active set. Defaults to TRUE and initializes the active set with those covariates that are most correlated with score residuals of an unconditional modFUN(update(formula, . ~ 1)).
m_max	maximum number of iterating the splicing algorithm.
m0	Transformation model for initialization
...	Currently ignored

Value

List containing the fitted model via mmlt, active set A and inactive set I.

---

abess_tram *Optimal subset selection for transformation models*

Description

Optimal subset selection for transformation models

Usage

```
abess_tram(
  formula,
  data,
  modFUN,
  supp,
  mandatory = NULL,
  k_max = supp,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  m0 = NULL,
  ...
)
```

Arguments

`formula` object of class `"formula"`.

`data` data frame containing the variables in the model.

`modFUN` function for fitting a transformation model, e.g., `BoxCox()`.

`supp` support size of the coefficient vector

`mandatory` formula of mandatory covariates, which will always be included and estimated in the model. Note that this also changes the intialization of the active set. The active set is then computed with regards to the model residuals of `modFUN(mandatory, ...)` instead of the unconditional model.

`k_max` maximum support size to consider during the splicing algorithm. Defaults to `supp`.

`thresh` threshold when to stop splicing. Defaults to 0.01 * `supp` * p * log(log(n)) / n$, where p denotes the number of predictors and n the sample size.

`init` initialize active set. Defaults to `TRUE` and initializes the active set with those covariates that are most correlated with score residuals of an unconditional `modFUN(update(formula, . ~ 1))`.

`m_max` maximum number of iterating the splicing algorithm.

`m0` Transformation model for initialization

`...` additional arguments supplied to `modFUN`.

Value

List containing the fitted model via `modFUN`, active set `A` and inactive set `I`.

Examples

```
set.seed(24101968)
library(tramvs)

N <- 1e2
P <- 5
nz <- 3
beta <- rep(c(1, 0), c(nz, P - nz))
X <- matrix(rnorm(N * P), nrow = N, ncol = P)
Y <- 1 + X %*% beta + rnorm(N)

dat <- data.frame(y = Y, x = X)

abess_tram(y ~ ., dat, modFUN = Lm, supp = 3)
```

AIC.tramvs *AIC "tramvs"*

Description

AIC "tramvs"

Usage

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

Arguments

object	object of class "tramvs"
...	additional arguments to AIC()

Value

Numeric vector containing AIC of best model

BoxCoxVS *Optimal subset selection in a BoxCox-type transformation model*

Description

Optimal subset selection in a BoxCox-type transformation model

Usage

```
BoxCoxVS(
  formula,
  data,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  parallel = FALSE,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

formula — object of class "formula".

data — data frame containing the variables in the model.

supp_max — maximum support which to call abess_tram with.

k_max — maximum support size to consider during the splicing algorithm. Defaults to supp.

thresh — threshold when to stop splicing. Defaults to 0.01 * supp * p * log(log(n)) / n$, where p denotes the number of predictors and n the sample size.

init — initialize active set. Defaults to TRUE and initializes the active set with those covariates that are most correlated with score residuals of an unconditional modFUN(update(formula, . ~ 1)).

m_max — maximum number of iterating the splicing algorithm.

parallel — toggle for parallel computing via future_lapply

future_args — arguments passed to plan; defaults to a "multisession" with supp_max workers

... — Additional arguments supplied to BoxCox

Value

See tramvs

coef.abess_tram *Coef "abess_tram"*

Description

Coef "abess_tram"

Usage

```
## S3 method for class 'abess_tram'
coef(object, ...)
```

Arguments

object — object of class "tramvs"

... — additional arguments to coef()

Value

Named numeric vector containing coefficient estimates see coef.tram

`coef.mmltvs` *Coef "mmltvs"*

Description

Coef "mmltvs"

Usage

```
## S3 method for class 'mmltvs'
coef(object, best_only = FALSE, ...)
```

Arguments

`object`	Object of class `"tramvs"`
`best_only`	Wether to return the coefficients of the best model only (default: FALSE)
`...`	additional arguments to `coef()`

Value

Vector (`best_only` = `TRUE`) or matrix (`best_only` = `FALSE`) of coefficients

`coef.tramvs` *Coef "tramvs"*

Description

Coef "tramvs"

Usage

```
## S3 method for class 'tramvs'
coef(object, best_only = FALSE, ...)
```

Arguments

`object`	Object of class `"tramvs"`
`best_only`	Wether to return the coefficients of the best model only (default: FALSE)
`...`	additional arguments to `coef()`

Value

Vector (`best_only` = `TRUE`) or matrix (`best_only` = `FALSE`) of coefficients

ColrVS *Optimal subset selection in a Colr-type transformation model*

Description

Optimal subset selection in a Colr-type transformation model

Usage

```
ColrVS(
  formula,
  data,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  parallel = FALSE,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

`formula` object of class `"formula"`.

`data` data frame containing the variables in the model.

`supp_max` maximum support which to call `abess_tram` with.

`k_max` maximum support size to consider during the splicing algorithm. Defaults to `supp`.

`thresh` threshold when to stop splicing. Defaults to 0.01 * supp * p * log(log(n)) / n$, where p denotes the number of predictors and n the sample size.

`init` initialize active set. Defaults to `TRUE` and initializes the active set with those covariates that are most correlated with score residuals of an unconditional `modFUN(update(formula, . ~ 1))`.

`m_max` maximum number of iterating the splicing algorithm.

`parallel` toggle for parallel computing via `future_lapply`

`future_args` arguments passed to `plan`; defaults to a `"multisession"` with `supp_max` workers

`...` Additional arguments supplied to `Colr`

Value

See `tramvs`

cor_init *Compute correlation for initializing the active set*

Description

Compute correlation for initializing the active set

Usage

```
cor_init(m0, mb)
```

Arguments

m0	modFUN(formula, data)
mb	modFUN(mandatory, data)

Value

Vector of correlations for initializing the active set, depends on type of model (see e.g. `cor_init.default`)

cor_init.default *Default method for computing correlation*

Description

Default method for computing correlation

Usage

```
## Default S3 method:
cor_init(m0, mb)
```

Arguments

m0	modFUN(formula, data)
mb	modFUN(mandatory, data)

Value

Vector of correlation for initializing the active set

cor_init.mmlt *Method for computing correlations in mmlts*

Description

Method for computing correlations in mmlts

Usage

```
## S3 method for class 'mmlt'
cor_init(m0, mb)
```

Arguments

m0	modFUN(formula, data)
mb	modFUN(mandatory, data)

Value

Vector of correlation for initializing the active set

cor_init.stram *Shit-scale tram method for computing correlation*

Description

Shit-scale tram method for computing correlation

Usage

```
## S3 method for class 'stram'
cor_init(m0, mb)
```

Arguments

m0	modFUN(formula, data)
mb	modFUN(mandatory, data)

Value

Vector of correlations for initializing the active set, includes both shift and scale residuals

cotramVS *Optimal subset selection in a cotram model*

Description

Optimal subset selection in a cotram model

Usage

```
cotramVS(
  formula,
  data,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  parallel = FALSE,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

`formula`	object of class `"formula"`.
`data`	data frame containing the variables in the model.
`supp_max`	maximum support which to call `abess_tram` with.
`k_max`	maximum support size to consider during the splicing algorithm. Defaults to `supp`.
`thresh`	threshold when to stop splicing. Defaults to 0.01 * `supp` * p * log(log(n)) / n$, where p denotes the number of predictors and n the sample size.
`init`	initialize active set. Defaults to `TRUE` and initializes the active set with those covariates that are most correlated with score residuals of an unconditional `modFUN(update(formula, . ~ 1))`.
`m_max`	maximum number of iterating the splicing algorithm.
`parallel`	toggle for parallel computing via `future_lapply`
`future_args`	arguments passed to `plan`; defaults to a `"multisession"` with `supp_max` workers
`...`	Additional arguments supplied to `cotram`

Value

See `tramvs`

CoxphVS *Optimal subset selection in a Coxph-type transformation model*

Description

Optimal subset selection in a Coxph-type transformation model

Usage

```
CoxphVS(
  formula,
  data,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  parallel = FALSE,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

formula	object of class "formula".
data	data frame containing the variables in the model.
supp_max	maximum support which to call abess_tram with.
k_max	maximum support size to consider during the splicing algorithm. Defaults to supp.
thresh	threshold when to stop splicing. Defaults to 0.01 * supp * p * log(log(n)) / n$, where p denotes the number of predictors and n the sample size.
init	initialize active set. Defaults to TRUE and initializes the active set with those covariates that are most correlated with score residuals of an unconditional modFUN(update(formula, . ~ 1)).
m_max	maximum number of iterating the splicing algorithm.
parallel	toggle for parallel computing via future_lapply
future_args	arguments passed to plan; defaults to a "multisession" with supp_max workers
...	Additional arguments supplied to Coxph

Value

See tramvs

LehmannVS *Optimal subset selection in a Lehmann-type transformation model*

Description

Optimal subset selection in a Lehmann-type transformation model

Usage

```
LehmannVS(
  formula,
  data,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  parallel = FALSE,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

formula	object of class "formula".
data	data frame containing the variables in the model.
supp_max	maximum support which to call abess_tram with.
k_max	maximum support size to consider during the splicing algorithm. Defaults to supp.
thresh	threshold when to stop splicing. Defaults to 0.01 * supp * p * log(log(n)) / n$, where p denotes the number of predictors and n the sample size.
init	initialize active set. Defaults to TRUE and initializes the active set with those covariates that are most correlated with score residuals of an unconditional modFUN(update(formula, . ~ 1)).
m_max	maximum number of iterating the splicing algorithm.
parallel	toggle for parallel computing via future_lapply
future_args	arguments passed to plan; defaults to a "multisession" with supp_max workers
...	Additional arguments supplied to Lehmann

Value

See tramvs

LmVS *Optimal subset selection in an Lm-type transformation model*

Description

Optimal subset selection in an Lm-type transformation model

Usage

```
LmVS(
  formula,
  data,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  parallel = FALSE,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

`formula` object of class `"formula"`.

`data` data frame containing the variables in the model.

`supp_max` maximum support which to call `abess_tram` with.

`k_max` maximum support size to consider during the splicing algorithm. Defaults to `supp`.

`thresh` threshold when to stop splicing. Defaults to 0.01 * supp * p * log(log(n)) / n$, where p denotes the number of predictors and n the sample size.

`init` initialize active set. Defaults to `TRUE` and initializes the active set with those covariates that are most correlated with score residuals of an unconditional `modFUN(update(formula, . ~ 1))`.

`m_max` maximum number of iterating the splicing algorithm.

`parallel` toggle for parallel computing via `future_lapply`

`future_args` arguments passed to `plan`; defaults to a `"multisession"` with `supp_max` workers

`...` Additional arguments supplied to `Lm`

Value

See `tramvs`

logLik.tramvs *logLik "tramvs"*

Description

logLik "tramvs"

Usage

```
## S3 method for class 'tramvs'
logLik(object, ...)
```

Arguments

`object`	object of class `"tramvs"`
`...`	additional arguments to `logLik()`

Value

Numeric vector containing log-likelihood of best model, see `logLik.tram`

mmltVS *Select optimal subset based on high dimensional BIC in mmlts*

Description

Select optimal subset based on high dimensional BIC in mmlts

Usage

```
mmltVS(
  mltargs,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  verbose = TRUE,
  parallel = FALSE,
  m0 = NULL,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

`mltargs` Arguments passed to `mmlt`

`supp_max` maximum support which to call `abess_tram` with.

`k_max` maximum support size to consider during the splicing algorithm. Defaults to `supp`.

`thresh` threshold when to stop splicing. Defaults to 0.01 * `supp` * p * log(log(n)) / n$, where p denotes the number of predictors and n the sample size.

`init` initialize active set. Defaults to `TRUE` and initializes the active set with those covariates that are most correlated with score residuals of an unconditional `modFUN(update(formula, . ~ 1))`.

`m_max` maximum number of iterating the splicing algorithm.

`verbose` show progress bar (default: `TRUE`)

`parallel` toggle for parallel computing via `future_lapply`

`m0` Transformation model for initialization

`future_args` arguments passed to `plan`; defaults to a `"multisession"` with `supp_max` workers

`...` Arguments passed on to `abess_mmlt`

`supp` support size of the coefficient vector

Details

L0-penalized (i.e., best subset selection) multivariate transformation models using the abess algorithm.

Value

object of class `"mltvs"`, containing the regularization path (information criterion `SIC` and coefficients `coefs`), the best fit (`best_fit`) and all other models (`all_fits`)

`plot.tramvs` *Plot "tramvs" object*

Description

Plot "tramvs" object

Usage

```
## S3 method for class 'tramvs'
plot(x, which = c("tune", "path"), ...)
```

Arguments

`x` object of class `"tramvs"`

`which` plotting either the regularization path (`"path"`) or the information criterion against the support size (`"tune"`, default)

`...` additional arguments to `plot()`

Value

Returns `invisible(NULL)`

`PolrVS` *Optimal subset selection in a Polr-type transformation model*

Description

Optimal subset selection in a Polr-type transformation model

Usage

```
PolrVS(
  formula,
  data,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  parallel = FALSE,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

`formula` object of class `"formula"`.

`data` data frame containing the variables in the model.

`supp_max` maximum support which to call `abess_tram` with.

`k_max` maximum support size to consider during the splicing algorithm. Defaults to `supp`.

`thresh` threshold when to stop splicing. Defaults to 0.01 * supp * p * log(log(n)) / n$, where p denotes the number of predictors and n the sample size.

`init` initialize active set. Defaults to `TRUE` and initializes the active set with those covariates that are most correlated with score residuals of an unconditional `modFUN(update(formula, . ~ 1))`.

`m_max`	maximum number of iterating the splicing algorithm.
`parallel`	toggle for parallel computing via `future_lapply`
`future_args`	arguments passed to `plan`; defaults to a `"multisession"` with `supp_max` workers
`...`	Additional arguments supplied to `Polr`

Value

See `tramvs`

`predict.tramvs` *Predict "tramvs"*

Description

Predict "tramvs"

Usage

```
## S3 method for class 'tramvs'
predict(object, ...)
```

Arguments

`object`	object of class `"tramvs"`
`...`	additional arguments to `predict.tram()`

Value

See `predict.tram`

`print.tramvs` *Print "tramvs"*

Description

Print "tramvs"

Usage

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

Arguments

`x`	object of class `"tramvs"`
`...`	ignored

Value

`"tramvs"` object is returned invisibly

`residuals.tramvs` *Residuals "tramvs"*

Description

Residuals "tramvs"

Usage

```
## S3 method for class 'tramvs'
residuals(object, ...)
```

Arguments

`object`	object of class `"tramvs"`
`...`	additional arguments to `residuals()`

Value

Numeric vector containing residuals of best model, see `residuals.tram`

`SIC` *SIC generic*

Description

SIC generic

Usage

```
SIC(object, ...)
```

Arguments

`object`	Model to compute SIC from
`...`	for methods compatibility only

Value

Numeric vector (`best_only` = `TRUE`) or data.frame with SIC values

SIC.tramvs *SIC "tramvs"*

Description

SIC "tramvs"

Usage

```
## S3 method for class 'tramvs'
SIC(object, best_only = FALSE, ...)
```

Arguments

`object`	object of class `"tramvs"`
`best_only`	Wether to return the coefficients of the best model only (default: FALSE)
`...`	for methods compatibility only

Value

Numeric vector (`best_only = TRUE`) or data.frame with SIC values

simulate.tramvs *Simulate "tramvs"*

Description

Simulate "tramvs"

Usage

```
## S3 method for class 'tramvs'
simulate(object, nsim = 1, seed = NULL, ...)
```

Arguments

`object`	object of class `"tramvs"`
`nsim`	number of simulations
`seed`	random seed for simulation
`...`	additional arguments to `simulate()`

Value

See `simulate.mlt`

summary.tramvs *Summary "tramvs"*

Description

Summary "tramvs"

Usage

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

Arguments

`object`	object of class `"tramvs"`
`...`	ignored

Value

`"tramvs"` object is returned invisibly

support.tramvs *Support "tramvs"*

Description

Support "tramvs"

Usage

```
## S3 method for class 'tramvs'
support(object, ...)
```

Arguments

`object`	object of class `"tramvs"`
`...`	ignored

Value

Character vector containing active set of best fit

SurvregVS *Optimal subset selection in a Survreg model*

Description

Optimal subset selection in a Survreg model

Usage

```
SurvregVS(
  formula,
  data,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  parallel = FALSE,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

formula	object of class "formula".
data	data frame containing the variables in the model.
supp_max	maximum support which to call abess_tram with.
k_max	maximum support size to consider during the splicing algorithm. Defaults to supp.
thresh	threshold when to stop splicing. Defaults to 0.01 * supp * p * log(log(n)) / n$, where p denotes the number of predictors and n the sample size.
init	initialize active set. Defaults to TRUE and initializes the active set with those covariates that are most correlated with score residuals of an unconditional modFUN(update(formula, . ~ 1)).
m_max	maximum number of iterating the splicing algorithm.
parallel	toggle for parallel computing via future_lapply
future_args	arguments passed to plan; defaults to a "multisession" with supp_max workers
...	Additional arguments supplied to Survreg

Value

See tramvs

tramvs — *Select optimal subset based on high dimensional BIC*

Description

Select optimal subset based on high dimensional BIC

Usage

```
tramvs(
  formula,
  data,
  modFUN,
  mandatory = NULL,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  m0 = NULL,
  verbose = TRUE,
  parallel = FALSE,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

`formula` object of class `"formula"`.

`data` data frame containing the variables in the model.

`modFUN` function for fitting a transformation model, e.g., `BoxCox()`.

`mandatory` formula of mandatory covariates, which will always be included and estimated in the model. Note that this also changes the intialization of the active set. The active set is then computed with regards to the model residuals of `modFUN(mandatory, ...)` instead of the unconditional model.

`supp_max` maximum support which to call `abess_tram` with.

`k_max` maximum support size to consider during the splicing algorithm. Defaults to `supp`.

`thresh` threshold when to stop splicing. Defaults to 0.01 * supp * p * log(log(n)) / n$, where p denotes the number of predictors and n the sample size.

`init` initialize active set. Defaults to `TRUE` and initializes the active set with those covariates that are most correlated with score residuals of an unconditional `modFUN(update(formula, . ~ 1))`.

`m_max` maximum number of iterating the splicing algorithm.

m0	Transformation model for initialization
verbose	show progress bar (default: TRUE)
parallel	toggle for parallel computing via future_lapply
future_args	arguments passed to plan; defaults to a "multisession" with supp_max workers
...	Arguments passed on to abess_tram
	supp support size of the coefficient vector

Details

L0-penalized (i.e., best subset selection) transformation models using the abess algorithm.

Value

object of class "tramvs", containing the regularization path (information criterion SIC and coefficients coefs), the best fit (best_fit) and all other models (all_fits)

Examples

```
set.seed(24101968)
library("tramvs")

N <- 1e2
P <- 5
nz <- 3
beta <- rep(c(1, 0), c(nz, P - nz))
X <- matrix(rnorm(N * P), nrow = N, ncol = P)
Y <- 1 + X %*% beta + rnorm(N)

dat <- data.frame(y = Y, x = X)
res <- tramvs(y ~ ., data = dat, modFUN = Lm)
plot(res, type = "b")
plot(res, which = "path")
```

Index