

Package ‘tramvs’

August 22, 2022

Type Package

Title Optimal Subset Selection for Transformation Models

Version 0.0-2

Description Greedy optimal subset selection for transformation models (Hothorn et al., 2018, <[doi:10.1111/sjos.12291](https://doi.org/10.1111/sjos.12291)>) based on the abess algorithm (Zhu et al., 2020, <[doi:10.1073/pnas.2014241117](https://doi.org/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

Suggests abess, tramnet, colorspace, knitr, mlt, TH.data, survival, ordinal

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.2

URL <http://ctm.R-forge.R-project.org>

VignetteBuilder knitr

NeedsCompilation no

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

Maintainer Lucas Kook <lucasheinrich.kook@uzh.ch>

Repository CRAN

Date/Publication 2022-08-22 09:20:03 UTC

R topics documented:

abess_tram	2
AIC.tramvs	4
BoxCoxVS	4
coef.abess_tram	5

coef.tramvs	6
ColrVS	6
cor_init	7
cor_init.default	8
cor_init.stram	8
cotramVS	9
CoxphVS	10
LehmannVS	11
LmVS	12
logLik.tramvs	13
plot.tramvs	13
PolrVS	14
predict.tramvs	15
print.tramvs	15
residuals.tramvs	16
SIC	16
SIC.tramvs	17
simulate.tramvs	17
summary.tramvs	18
support.tramvs	18
SurvregVS	19
tramvs	20

Index	22
--------------	-----------

abess_tram	<i>Optimal subset selection for transformation models</i>
------------	---

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 initialization 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 $0.01 * \text{supp} * p * \log(\log(n)) / n\$$, 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()

ValueNumeric 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,  
  ...  
)
```

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 $0.01 * \text{supp} * p * \log(\log(n)) / n$, 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 <code>modFUN(update(formula, . ~ 1))</code> .
m_max	maximum number of iterating the splicing algorithm.
...	Additional arguments supplied to BoxCox

Value

See [tramvs](#)

coef.abess_tram	<i>Coef "abess_tram"</i>
-----------------	--------------------------

Description

Coef "abess_tram"

Usage

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

Arguments

object	object of class "tramvs"
...	additional arguments to <code>coef()</code>

Value

Named numeric vector containing coefficient estimates see [coef.tram](#)

<code>coef.tramvs</code>	<i>Coef "tramvs"</i>
--------------------------	----------------------

Description

Coef "tramvs"

Usage

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

Arguments

<code>object</code>	Object of class "tramvs"
<code>best_only</code>	Whether to return the coefficients of the best model only (default: FALSE)
<code>...</code>	additional arguments to <code>coef()</code>

Value

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

<code>ColrVS</code>	<i>Optimal subset selection in a Colr-type transformation model</i>
---------------------	---

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,
  ...
)
```

Arguments

formula	object of class "formula".
data	data frame containing the variables in the model.
supp_max	maximum support which to call <code>abess_tram</code> with.
k_max	maximum support size to consider during the splicing algorithm. Defaults to <code>supp</code> .
thresh	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n$, 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 co-variables that are most correlated with score residuals of an unconditional <code>modFUN(update(formula, . ~ 1))</code> .
m_max	maximum number of iterating the splicing algorithm.
...	Additional arguments supplied to <code>Colr</code>

Value

See [tramvs](#)

cor_init	<i>Compute correlation for initializing the active set</i>
----------	--

Description

Compute correlation for initializing the active set

Usage

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

Arguments

m0	<code>modFUN(formula, data)</code>
mb	<code>modFUN(mandatory, data)</code>

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.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,  
  ...  
)
```

Arguments

<code>formula</code>	object of class "formula".
<code>data</code>	data frame containing the variables in the model.
<code>supp_max</code>	maximum support which to call <code>abess_tram</code> with.
<code>k_max</code>	maximum support size to consider during the splicing algorithm. Defaults to <code>supp</code> .
<code>thresh</code>	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n\$$, where p denotes the number of predictors and n the sample size.
<code>init</code>	initialize active set. Defaults to <code>TRUE</code> and initializes the active set with those co-variables that are most correlated with score residuals of an unconditional <code>modFUN(update(formula, . ~ 1))</code> .
<code>m_max</code>	maximum number of iterating the splicing algorithm.
<code>...</code>	Additional arguments supplied to <code>cotram</code>

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,
  ...
)
```

Arguments

<code>formula</code>	object of class "formula".
<code>data</code>	data frame containing the variables in the model.
<code>supp_max</code>	maximum support which to call <code>abess_tram</code> with.
<code>k_max</code>	maximum support size to consider during the splicing algorithm. Defaults to <code>supp</code> .
<code>thresh</code>	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n$, where p denotes the number of predictors and n the sample size.
<code>init</code>	initialize active set. Defaults to <code>TRUE</code> and initializes the active set with those co-variables that are most correlated with score residuals of an unconditional <code>modFUN(update(formula, . ~ 1))</code> .
<code>m_max</code>	maximum number of iterating the splicing algorithm.
<code>...</code>	Additional arguments supplied to Coxph

Value

See [tramvs](#)

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,
  ...
)
```

Arguments

<code>formula</code>	object of class "formula".
<code>data</code>	data frame containing the variables in the model.
<code>supp_max</code>	maximum support which to call <code>abess_tram</code> with.
<code>k_max</code>	maximum support size to consider during the splicing algorithm. Defaults to <code>supp</code> .
<code>thresh</code>	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n\$$, where p denotes the number of predictors and n the sample size.
<code>init</code>	initialize active set. Defaults to <code>TRUE</code> and initializes the active set with those co-variables that are most correlated with score residuals of an unconditional <code>modFUN(update(formula, . ~ 1))</code> .
<code>m_max</code>	maximum number of iterating the splicing algorithm.
<code>...</code>	Additional arguments supplied to <code>Lehmann</code>

Value

See [tramvs](#)

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,
  ...
)
```

Arguments

<code>formula</code>	object of class "formula".
<code>data</code>	data frame containing the variables in the model.
<code>supp_max</code>	maximum support which to call <code>abess_tram</code> with.
<code>k_max</code>	maximum support size to consider during the splicing algorithm. Defaults to <code>supp</code> .
<code>thresh</code>	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n$, where p denotes the number of predictors and n the sample size.
<code>init</code>	initialize active set. Defaults to <code>TRUE</code> and initializes the active set with those co-variables that are most correlated with score residuals of an unconditional <code>modFUN(update(formula, . ~ 1))</code> .
<code>m_max</code>	maximum number of iterating the splicing algorithm.
<code>...</code>	Additional arguments supplied to <code>Lm</code>

Value

See [tramvs](#)

logLik.tramvs	<i>logLik "tramvs"</i>
---------------	------------------------

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](#)

plot.tramvs	<i>Plot "tramvs" object</i>
-------------	-----------------------------

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,
  ...
)
```

Arguments

<code>formula</code>	object of class "formula".
<code>data</code>	data frame containing the variables in the model.
<code>supp_max</code>	maximum support which to call <code>abess_tram</code> with.
<code>k_max</code>	maximum support size to consider during the splicing algorithm. Defaults to <code>supp</code> .
<code>thresh</code>	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n\$$, where p denotes the number of predictors and n the sample size.
<code>init</code>	initialize active set. Defaults to <code>TRUE</code> and initializes the active set with those co-variables that are most correlated with score residuals of an unconditional <code>modFUN(update(formula, . ~ 1))</code> .
<code>m_max</code>	maximum number of iterating the splicing algorithm.
<code>...</code>	Additional arguments supplied to Polr

Value

See [tramvs](#)

predict.tramvs	<i>Predict "tramvs"</i>
----------------	-------------------------

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	<i>Print "tramvs"</i>
--------------	-----------------------

Description

Print "tramvs"

Usage

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

Arguments

x	object of class "tramvs"
...	ignored

Value

"tramvs" object is returned invisibly

<code>residuals.tramvs</code>	<i>Residuals "tramvs"</i>
-------------------------------	---------------------------

Description

Residuals "tramvs"

Usage

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

Arguments

<code>object</code>	object of class "tramvs"
<code>...</code>	additional arguments to <code>residuals()</code>

Value

Numeric vector containing residuals of best model, see [residuals.tram](#)

<code>SIC</code>	<i>SIC generic</i>
------------------	--------------------

Description

SIC generic

Usage

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

Arguments

<code>object</code>	Model to compute SIC from
<code>...</code>	for methods compatibility only

Value

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

SIC.tramvs	<i>SIC "tramvs"</i>
------------	---------------------

Description

SIC "tramvs"

Usage

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

Arguments

object	object of class "tramvs"
best_only	Whether 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	<i>Simulate "tramvs"</i>
-----------------	--------------------------

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

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,
  ...
)
```

Arguments

<code>formula</code>	object of class "formula".
<code>data</code>	data frame containing the variables in the model.
<code>supp_max</code>	maximum support which to call <code>abess_tram</code> with.
<code>k_max</code>	maximum support size to consider during the splicing algorithm. Defaults to <code>supp</code> .
<code>thresh</code>	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n\$$, where p denotes the number of predictors and n the sample size.
<code>init</code>	initialize active set. Defaults to <code>TRUE</code> and initializes the active set with those co-variables that are most correlated with score residuals of an unconditional <code>modFUN(update(formula, . ~ 1))</code> .
<code>m_max</code>	maximum number of iterating the splicing algorithm.
<code>...</code>	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,
  ...
)
```

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 initialization 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 $0.01 * \text{supp} * p * \log(\log(n)) / n$, 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.

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 = "s")
plot(res, which = "path")
```

Index

abess_tram, 2
AIC.tramvs, 4

BoxCox, 5
BoxCoxVS, 4

coef.abess_tram, 5
coef.tram, 5
coef.tramvs, 6
Colr, 7
ColrVS, 6
cor_init, 7
cor_init.default, 7, 8
cor_init.stram, 8
cotram, 9
cotramVS, 9
Coxph, 10
CoxphVS, 10

LehmannVS, 11
Lm, 12
LmVS, 12
logLik.tram, 13
logLik.tramvs, 13

plot.tramvs, 13
Polr, 14
PolrVS, 14
predict.tram, 15
predict.tramvs, 15
print.tramvs, 15

residuals.tram, 16
residuals.tramvs, 16

SIC, 16
SIC.tramvs, 17
simulate.mlt, 17
simulate.tramvs, 17
summary.tramvs, 18
support.tramvs, 18

Survreg, 19
SurvregVS, 19

tramvs, 5, 7, 9–12, 14, 19, 20