Package 'S3VS'

January 13, 2026

Type Package

Title Structured Screen-and-Select Variable Selection in Linear, Generalized Linear, and Survival Models

Version 1.0

Date 2025-12-30

Maintainer Nilotpal Sanyal <nsanyal@utep.edu>

Description Performs variable selection using the structured screen-and-select (S3VS) framework in linear models, generalized linear models with binary data, and survival models such as the Cox model and accelerated failure time (AFT) model.

License GPL (>= 2)

Imports glmnet, ncvreg, survival, mombf, future.apply, eha, pec, aftgee, afthd

Suggests rjags, knitr, doParallel

VignetteBuilder knitr

NeedsCompilation no

Author Nilotpal Sanyal [aut, cre], Padmore N. Prempeh [aut]

Repository CRAN

Date/Publication 2026-01-13 18:00:14 UTC

Contents

`S3VS-package` *Structured Screen-and-Select Variable Selection in Linear, Generalized Linear, and Survival Models*

Description

Performs variable selection using the structured screen-and-select (S3VS) framework in linear models, generalized linear models with binary data, and survival models such as the Cox model and accelerated failure time (AFT) model.

Details

The **S3VS** package implements the Structured Screen-and-Select Variable Selection (S3VS) framework for linear models, generalized linear models with binary responses, and survival models (Cox proportional hazards and accelerated failure time models).

The central entry point is `S3VS`, which dispatches to a family-specific routine via the argument `family`:

- `S3VS_LM` for linear models,
- `S3VS_GLM` for generalized linear models with binary outcomes,
- `S3VS_SURV` for survival models.

The S3VS workflow proceeds through the following steps, each handled by helper functions:

Stopping rule check `looprun` determines whether the iterative screen-and-select process should continue.

Leading variable identification `get_leadvars` identifies leading variables; family-specific versions are `get_leadvars_LM`, `get_leadvars_GLM`, and `get_leadvars_SURV`.

Leading set identification `get_leadsets` identifies the leading set for each leading variable.

Selection within leading sets `VS_method` performs selection within leading sets; family-specific methods include `VS_method_LM`, `VS_method_GLM`, `VS_method_SURV`, and `bridge_aft` implements BRIDGE specifically for AFT models.

Aggregation of selected variables `select_vars` retains promising variables as selected from an iteration.

Aggregation of non-selected variables (optional) `remove_vars` removes variables deemed uninformative from future iterations (if no variable is selected in the current iteration by `select_vars`).

Response update (optional) `update_y` enables iterative response updates; family-specific variants include `update_y_LM` and `update_y_GLM`.

Together, these functions form a structured, iterative pipeline for efficient variable screening and selection in high-dimensional regression and survival analysis.

Prediction `pred_S3VS` produces predictions using variables selected by S3VS, calling `pred_S3VS_LM`, `pred_S3VS_GLM`, or `pred_S3VS_SURV` as appropriate.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

Maintainer: Nilotpal Sanyal <nsanyal@utep.edu>

`bridge_aft` *Bridge-Penalized AFT Regression via Iteratively Reweighted LASSO*

Description

`bridge_aft` fits an accelerated failure time (AFT) model using an iterative reweighted LASSO scheme to approximate a bridge (L_γ) penalty on the regression coefficients.

Usage

```
bridge_aft(y, X, gamma = 0.5, alpha = 1, max_iter = 100, tol = 1e-05)
```

Arguments

`y`	Response; a list of two elements `time` and `status`.
`X`	Predictor matrix. Can be a base matrix or something `as.matrix()` can coerce. No missing values are allowed.
`gamma`	Bridge penalty exponent γ (default `0.5`). Values in $(0, 1]$ approximate nonconvex penalties; $\gamma = 1$ reduces to LASSO-like weighting.
`alpha`	Elastic-net mixing parameter passed to `glmnet::cv.glmnet` (default `1` for LASSO; `0` is ridge; `(0,1)` is elastic net).
`max_iter`	Maximum number of outer reweighting iterations (default `100`).
`tol`	Convergence tolerance on the L_1 change in coefficients between iterations (default `1e-5`).

Value

A list with components:

`beta`	Numeric vector of estimated coefficients of length p.
`gamma`	The bridge exponent used in the fit.
`iterations`	Number of outer reweighting iterations performed.

Author(s)

Padmore Prempeh <pprempeh@albany.edu>, Nilotpal Sanyal <nsanyal@utep.edu>

References

Jian Huang and Shuangge Ma. Variable selection in the accelerated failure time model via the bridge method. Lifetime Data Analysis, 16(2):176-195, 2010.

See Also

`glmnet`, `cv.glmnet`, `Surv`, `survfit`

Examples

```
set.seed(1)
n  <- 50
p  <- 10
X  <- matrix(rnorm(n * p), n, p)
beta_true <- c(runif(10, -1.5, 1.5), rep(0, p - 10))
linpred   <- as.vector(X %*% beta_true)

## Generate log-normal AFT survival times (no censoring in this simple example)
sigma <- 0.6
logT  <- linpred + rnorm(n, sd = sigma)
time  <- exp(logT)
delta <- rep(1, n)  # all events (censoring ignored by current implementation)

y_surv <- list(time = time, status = delta)
fit <- bridge_aft(y_surv, X, gamma = 0.5, alpha = 1, max_iter = 50, tol = 1e-5)
str(fit)
fit$beta[1:10]
```

`get_leadsets` *Identify Leading Sets of Covariates via Inter-Predictor Associations*

Description

`get_leadsets` identifies, for a specified *leading variable*, a set of associated predictors, the leading set, based on inter-predictor associations (absolute value of the correlation coefficient).

Usage

```
get_leadsets(x_lead, X, method = c("topk", "fixedthresh", "percthresh"), param)
```

Arguments

x_lead	Vector with values of the *leading variable*
X	Predictor matrix. Must contain the *leading variable*. Can be a base matrix or something as.matrix() can coerce. No missing values are allowed.
method	Rule for constructing, for each *leading variable*, the set of associated predictors (the "leading set") using inter-predictor association (absolute value of the correlation coefficient); one of c("topk", "fixedthresh", "percthresh"). "topk" keeps the predictors with the largest k association values; "fixedthresh" keeps predictors whose association is greater than or equal to a specified threshold; "percthresh" keeps predictors whose association is within a given percentage of the best.
param	Tuning parameter for method; If "topk", supply an integer k (keep the top k). If "fixedthresh", supply a numeric threshold (keep predictors with association $\geq$ threshold). If "percthresh", supply a percentage in $(0, 100]$ (keep predictors with association $\geq$ that percent of the highest association).

Value

A character vector containing the names of the predictors.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

Examples

```
# Simulate continuous data
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
y <- X[,1] + 0.5 * X[,2] + rnorm(n)
leadvars <- get_leadvars_LM(y = y, X = X, method = "topk", param = list(k=2))
get_leadsets(X[,leadvars[1]], X, method = "percthresh", param = list(thresh = 0.2))
```

---

get_leadvars *Screening Predictors As 'Leading Variables' By Evaluating Predictor-Response Associations*

Description

get_leadvars screens some predictors as "leading variables" based on predictor-response associations in linear, generalized linear, and survival models.

Usage

```
get_leadvars(y, X, family = c("normal","binomial","survival"),
  surv_model = c("AFT", "COX"),
  method = c("topk", "fixedthresh", "percthresh"), param,
  varsselected = NULL, varsleft = colnames(X), parallel = FALSE)
```

Arguments

y	Response. If family = "normal", a numeric vector. If family = "binomial", a numeric/integer/logical vector with values in {0,1}. If family = "survival", a list with components time and status (1 = event, 0 = censored).
X	Predictor matrix. Can be a base matrix or something as.matrix() can coerce. No missing values are allowed.
family	Model family; one of c("normal","binomial","survival"). Determines which engine is called (get_leadvars_LM, get_leadvars_GLM, or get_leadvars_SURV).
surv_model	Character string specifying the survival model (family="survival" only). Must be explicitly provided; there is no default. Values are "Cox" for proportional hazards models, "AFT" for accelerated failure time models.
method	Screening rule, one of c("topk", "fixedthresh", "percthresh"). The association measure depends on family (e.g., correlation for "normal", eta-squared for "binomial", or marginal utility for "survival"). "topk" keeps the predictors with the largest k association values; "fixedthresh" keeps predictors whose association is greater than or equal to a specified threshold; "percthresh" keeps predictors whose association is within a given percentage of the best.
param	Tuning parameter for method. If "topk", supply an integer k (keep the top k). If "fixedthresh", supply a numeric threshold (keep predictors with association $\geq$ threshold). If "percthresh", supply a percentage in $(0, 100]$ (keep predictors with association $\geq$ that percent of the highest association).
varsselected	Used only when family=survival. A character vector containing the predictors that are already selected in previous iterations. The association measure, conditional utility, is computed controling for these predictors. NULL, by default.
varsleft	Used only when family=survival. A character vector containing the predictors that are neither selected, nor removed from consideration in previous iterations. Leading predictors are chosen from these predictors. colnames(X), by default.
parallel	Logical. If TRUE, attempts to perform some computations in parallel mode in binomial and survival families, which is strongly recommended for faster execution. Defaults to FALSE.

Value

A character vector containing the names of the *leading variables*.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

See Also

get_leadvars_LM, get_leadvars_GLM, get_leadvars_SURV,

Examples

```
# Simulate continuous data
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
y <- X[,1] + 0.5 * X[,2] + rnorm(n)
# Select leading variables
leadvars <- get_leadvars(y = y, X = X, family = "normal",
                         method = "topk", param = list(k=2))
leadvars
```

---

get_leadvars_GLM *Screening Predictors As 'Leading Variables' By Evaluating Predictor-Response Associations In Generalized Linear Models*

Description

get_leadvars_GLM screens some predictors as "leading variables" based on predictor-response associations in generalized linear models.

Usage

```
get_leadvars_GLM(y, X, method = c("topk", "fixedetasqthresh", "percetasqthresh"), param)
```

Arguments

y	Response. A numeric/integer/logical vector with values in {0,1}.
X	Predictor matrix. Can be a base matrix or something as.matrix() can coerce. No missing values are allowed.
method	Screening rule, one of c("topk", "fixedthresh", "percthresh"). The association measure is eta-squared. "topk" keeps the predictors with the largest k association values; "fixedthresh" keeps predictors whose association is greater than or equal to a specified threshold; "percthresh" keeps predictors whose association is within a given percentage of the best.
param	Tuning parameter for method. If "topk", supply an integer k (keep the top k). If "fixedthresh", supply a numeric threshold (keep predictors with association $\geq$ threshold). If "percthresh", supply a percentage in $(0, 100]$ (keep predictors with association $\geq$ that percent of the highest association).

Value

A character vector containing the names of the leading varibales.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

Examples

```
# Simulate binary data
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
eta <- X[,1] + 0.5 * X[,2]
prob <- 1 / (1 + exp(-eta))
y <- rbinom(n, size = 1, prob = prob)
# Select leading variables
leadvars <- get_leadvars_GLM(y = y, X = X, method = "topk", param = list(k=2))
leadvars
```

`get_leadvars_LM` *Screening Predictors As 'Leading Variables' By Evaluating Predictor-Response Associations In Linear Models*

Description

`get_leadvars_LM` screens some predictors as "leading variables" based on predictor-response associations in linear models.

Usage

```
get_leadvars_LM(y, X, method = c("topk", "fixedcorthresh", "perccorthresh"), param)
```

Arguments

`y`	Response. A numeric vector.
`X`	Predictor matrix. Can be a base matrix or something `as.matrix()` can coerce. No missing values are allowed.
`method`	Screening rule, one of `c("topk", "fixedthresh", "percthresh")`. The association measure is correlation. `"topk"` keeps the predictors with the largest k association values; `"fixedthresh"` keeps predictors whose association is greater than or equal to a specified threshold; `"percthresh"` keeps predictors whose association is within a given percentage of the best.
`param`	Tuning parameter for `method`. If `"topk"`, supply an integer k (keep the top k). If `"fixedthresh"`, supply a numeric threshold (keep predictors with association $\geq$ threshold). If `"percthresh"`, supply a percentage in $(0, 100]$ (keep predictors with association $\geq$ that percent of the highest association).).

Value

A character vector containing the names of the leading varibales.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

Examples

```
# Simulate continuous data
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
y <- X[,1] + 0.5 * X[,2] + rnorm(n)
# Select leading variables
leadvars <- get_leadvars_LM(y = y, X = X, method = "topk", param = list(k=2))
leadvars
```

---

get_leadvars_SURV *Screening Predictors As "Leading Variables" By Evaluating Predictor-Response Associations In Survival Models*

Description

`get_leadvars_SURV` screens some predictors as "leading variables" based on predictor-response associations in survival models.

Usage

```
get_leadvars_SURV(y, X, surv_model = c("AFT", "COX"),
  method = c("topk", "fixedmuthresh", "percmuthresh"), param,
  varsselected = NULL, varsleft = colnames(X), parallel = FALSE)
```

Arguments

`y`	Response. A list with components `time` and `status` (1 = event, 0 = censored).
`X`	Predictor matrix. Can be a base matrix or something `as.matrix()` can coerce. No missing values are allowed.
`surv_model`	Character string specifying the survival model. Must be explicitly provided; there is no default. Values are `"Cox"` for proportional hazards models, `"AFT"` for accelerated failure time models.
`method`	Screening rule, one of `c("topk", "fixedthresh", "percthresh")`. The association measure is marginal utility. `"topk"` keeps the predictors with the largest k association values; `"fixedthresh"` keeps predictors whose association is greater than or equal to a specified threshold; `"percthresh"` keeps predictors whose association is within a given percentage of the best.

param	Tuning parameter for method. If "topk", supply an integer k (keep the top k). If "fixedthresh", supply a numeric threshold (keep predictors with association $\geq$ threshold). If "percthresh", supply a percentage in $(0, 100]$ (keep predictors with association $\geq$ that percent of the highest association).
varsselected	A character vector containing the predictors that are already selected in previous iterations. The association measure, conditional utility, is computed controling for these predictors. NULL, by default.
varsleft	A character vector containing the predictors that are neither selected, nor removed from consideration in previous iterations. Leading predictors are chosen from these predictors.
parallel	Logical. If TRUE, attempts to perform some computations in parallel mode in binomial and survival families, which is strongly recommended for faster execution. Defaults to colnames(X).

Value

A character vector containing the names of the leading varibales.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

Examples

```
# Simulate survival data (Cox)
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
eta <- X[,1] + 0.5 * X[,2]
base_rate <- 0.05
T_event <- rexp(n, rate = base_rate * exp(eta))
C <- rexp(n, rate = 0.03)
time <- pmin(T_event, C)
status <- as.integer(T_event <= C)
y_surv <- list(time = time, status = status)
# Select leading variables
leadvars <- get_leadvars_SURV(y = y_surv, X = X, surv_model = "COX",
                              method = "topk", param = list(k=2),
                              varsselected = NULL, varsleft = colnames(X))
leadvars
```

---

looprun *Decide Whether to Run Another S3VS Iteration*

Description

looprun evaluates simple stopping criteria for the S3VS procedure and returns an indicator of whether *one more iteration* should be executed.

Usage

```
looprun(varsselected, varsleft, max_nocollect, m, nskip)
```

Arguments

varsselected	Character vector with names of predictors selected so far. Only its length is used; NULL is treated as length 0.
varsleft	Character vector with names of candidate predictors that remain available for selection in future iterations. Only its length is used; NULL is treated as length 0.
max_nocollect	Integer count of iterations *up to now* in which no new predictors were selected.
m	Maximum allowed number of selected predictors (target cap for length(varsselected)).
nskip	Maximum allowed number of "no-collection" iterations before stopping.

Details

An additional S3VS iteration is recommended *iff* all three conditions hold:

$$|\texttt{varsselected}| < m,$$

$$|\texttt{varsleft}| > 0,$$

$$\texttt{max\_nocollect} < \texttt{nskip}.$$

Value

1 if another iteration should run, 0 otherwise.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

Examples

```
looprun(varsselected = c("x1","x2","x3"),
        varsleft     = paste0("x", 4:23),
        max_nocollect = 0,
        m = 10,
        nskip = 2)
```

pred_S3VS *Prediction Using S3VS-Selected Predictors*

Description

pred_S3VS performs prediction using predictors selected by S3VS in linear, generalized linear, and survival models.

Usage

```
pred_S3VS(y, X, family, surv_model = NULL, method)
```

Arguments

y	Response. If family = "normal", a numeric vector. If family = "binomial", a numeric/integer/logical vector with values in {0,1}. If family = "survival", a list with components time and status (1 = event, 0 = censored).
X	Predictor matrix. This should include predictors selected by S3VS. Can be a base matrix or something as.matrix() can coerce. No missing values are allowed.
family	Model family; one of c("normal","binomial","survival"). Determines which engine is called (pred_LM, pred_GLM, or pred_SURV).
surv_model	Character string specifying the survival model (family="survival" only). Must be explicitly provided; there is no default. Values are "Cox" for proportional hazards models, "AFT" for accelerated failure time models.
method	Character string indicating the prediction method used. Allowed values depend on family: for family = "normal"(functions pred_S3VS_LM), available options are "NLP", "LASSO", "SCAD", "MCP"; for "binomial" (S3VS_GLM), available options are "NLP", "LASSO"; for family = "survival" (S3VS_SURV), available options are "COXGLMNET" for surv_model = "COX" and .... for surv_model = "AFT". See Details for more information.

Value

A list containing:

y.pred	Predicted response
coef	Coefficient estimates of the predictors used for prediction

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

See Also

pred_S3VS_LM, pred_S3VS_GLM, pred_S3VS_SURV

Examples

```
# Simulate continuous data
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
y <- X[,1] + 0.5 * X[,2] + rnorm(n)
# Run S3VS for LM
res_lm <- S3VS(y = y, X = X, family = "normal",
               method_xy = "topk", param_xy = list(k=1),
               method_xx = "topk", param_xx = list(k=3),
               vsel_method = "LASSO", method_sel = "conservative",
               method_rem = "conservative_begin", rem_regout = FALSE,
               m = 100, nskip = 3, verbose = TRUE, seed = 123)
pred_lm <- pred_S3VS(y = y, X = X[,res_lm$selected], family = "normal", method = "LASSO")
```

---

pred_S3VS_GLM *Prediction Using S3VS-Selected Predictors in Survival Models*

Description

pred_S3VS performs prediction using predictors selected by S3VS in survival models.

Usage

```
pred_S3VS_GLM(y, X, method = c("NLP", "LASSO"))
```

Arguments

y	Response. A numeric/integer/logical vector with values in {0,1}.
X	Predictor matrix. This should include predictors selected by S3VS. Can be a base matrix or something as.matrix() can coerce. No missing values are allowed.
method	Character string indicating the prediction method used. Available options are "NLP", "LASSO".

Value

A list containing:

y.pred	Predicted response
coef	Coefficient estimates of the predictors used for prediction

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

See Also

rnlp, cv.glmnet

Examples

```
# Simulate binary data
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
eta <- X[,1] + 0.5 * X[,2]
prob <- 1 / (1 + exp(-eta))
y <- rbinom(n, size = 1, prob = prob)
# Predict
pred_glm <- pred_S3VS_GLM(y = y, X = X[,1:3], method = "LASSO")
pred_glm
```

---

pred_S3VS_LM *Prediction Using S3VS-Selected Predictors in Linear Models*

Description

pred_S3VS performs prediction using predictors selected by S3VS in linear models.

Usage

```
pred_S3VS_LM(y, X, method)
```

Arguments

y	Response. A numeric vector.
X	Predictor matrix. This should include predictors selected by S3VS. Can be a base matrix or something as.matrix() can coerce. No missing values are allowed.
method	Character string indicating the prediction method used. Available options are "NLP", "LASSO", "SCAD", "MCP"

Value

A list containing:

y.pred	Predicted response
coef	Coefficient estimates of the predictors used for prediction

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

See Also

rnlp, cv.glmnet, cv.ncvreg

Examples

```
# Simulate continuous data
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
y <- X[,1] + 0.5 * X[,2] + rnorm(n)
# Run S3VS for LM
res_lm <- S3VS(y = y, X = X, family = "normal",
               method_xy = "topk", param_xy = list(k=1),
               method_xx = "topk", param_xx = list(k=3),
               vsel_method = "LASSO", method_sel = "conservative",
               method_rem = "conservative_begin", rem_regout = FALSE,
               m = 100, nskip = 3, verbose = TRUE, seed = 123)
pred_lm <- pred_S3VS_LM(y = y, X = X[,res_lm$selected], method = "LASSO")
pred_lm
```

---

pred_S3VS_SURV *Predicted Survival Probabilities Using S3VS-Selected Predictors in Generalized Linear Models*

Description

pred_S3VS returns predicted survival probabilities using predictors selected by S3VS in generalized linear models.

Usage

```
pred_S3VS_SURV(y, X, surv_model = c("AFT", "COX"), method = c("AFTREG", "AFTGEE"), times)
```

Arguments

y	Response. A list with components time and status (1 = event, 0 = censored).
X	Predictor matrix. This should include predictors selected by S3VS. Can be a base matrix or something as.matrix() can coerce. No missing values are allowed.
surv_model	Character string specifying the survival model. Must be explicitly provided; there is no default. Values are "Cox" for proportional hazards models, "AFT" for accelerated failure time models.
method	Character string indicating the prediction method used. Available options are "COXGLMNET" for surv_model = "COX" and "AFTREG" and "AFTGEE" for surv_model = "AFT". See Details for more information.
times	Vector of time points where predicted survival probabilities will be computed.

Value

A list containing:

y.pred	Predicted response
coef	Coefficient estimates of the predictors used for prediction

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

See Also

cv.glmnet, coxph, aftreg, aftgee

Examples

```
# Simulate survival data (Cox)
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
eta <- X[,1] + 0.5 * X[,2]
base_rate <- 0.05
T_event <- rexp(n, rate = base_rate * exp(eta))
C <- rexp(n, rate = 0.03)
time <- pmin(T_event, C)
status <- as.integer(T_event <= C)
y_surv <- list(time = time, status = status)
# Run S3VS for linear models
res_surv <- S3VS(y = y_surv, X = X, family = "survival",
                 surv_model = "COX", vsel_method = "COXGLMNET",
                 method_xy = "topk", param_xy = list(k = 1),
                 method_xx = "topk", param_xx = list(k = 3),
                 method_sel = "conservative", method_rem = "conservative_begin",
                 sel_regout = FALSE, rem_regout = FALSE,
                 m = 100, nskip = 3, verbose = TRUE, seed = 123)
pred_surv <- pred_S3VS_SURV(y = y_surv, X = X[,res_surv$selected],
                            surv_model = "COX", method = "COXGLMNET")
pred_surv
```

---

remove_vars *Aggregate Not-Selected Predictors for Removal Across Multiple Leading Sets*

Description

remove_vars combines lists of predictors that were *not selected* from multiple leading sets into a single set to remove, using either a liberal (union) rule or a conservative (progressive intersection) rule.

Usage

```
remove_vars(listnotselect,
  method = c("conservative_begin", "conservative_end", "liberal"))
```

Arguments

`listnotselect`	A `list` of vectors, each containing names of the predictors not selected in the corresponding leading set.
`method`	Aggregation rule; one of `"conservative_begin"`, `"conservative_end"`, or `"liberal"`. Referring to the sets (vectors included in `listselect`) of not-selected predictors:

`"liberal"` Returns the *union* of all sets: `unique(unlist(listnotselect))`.

`"conservative_begin"` Returns the *last non-empty intersection* when intersecting the **first** $i = 1, 2, \ldots$ sets in order (note that, the first set is assumed to be non-empty, because that will automatically be true if `remove_vars` is being called by `S3VS` or its family-specific engines). The procedure stops once the running intersection becomes empty and returns the previous (last non-empty) intersection. Order of `listnotselect` matters.

`"conservative_end"` Returns the *last non-empty intersection* when intersecting the **last** $i = 1, 2, \ldots$ sets in order (if the last set is empty, the function finds the first non-empty set from the end, and then, starts the intersection process from that set). The procedure stops once the running intersection becomes empty and returns the previous (last non-empty) intersection. Order of `listnotselect` matters.

Details

The liberal rule favors inclusiveness (drop all predictors that were not selected in an iteration), whereas the conservative rule favors stability across earlier/latter leading sets (drop only predictors consistently absent in earlier/latter leading sets).

Value

Vector with names of the predictors that are not selected till the current S3VS iteration and to be removed from all future iterations.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

Examples

```
listselect <- list(
  c("V1","V2","V23"),
  c("V4","V2","V23"),
  c("V4","V5","V23")
)
remove_vars(listselect, method="liberal")
```

S3VS *Structured Screen-and-Select Variable Selection in Linear, Generalized Linear, and Survival Models*

Description

S3VS is the main function that performs variable selection based on the structured screen-and-select framework in linear, generalized linear, and survival models.

Usage

```
S3VS(
  y,
  X,
  family = c("normal", "binomial", "survival"),
  cor_xy = NULL,
  surv_model = c("COX", "AFT"),
  method_xy = c("topk", "fixedthresh", "percthresh"),
  param_xy,
  method_xx = c("topk", "fixedthresh", "percthresh"),
  param_xx,
  vsel_method = NULL,
  alpha = 0.5,
  method_sel = c("conservative", "liberal"),
  method_rem = c("conservative_begin", "conservative_end", "liberal"),
  sel_regout = FALSE,
  rem_regout = FALSE,
  update_y_thresh = 0.5,
  m = 100,
  nskip = 3,
  verbose = FALSE,
  seed = NULL,
  parallel = FALSE
)
```

Arguments

y	Response. If family = "normal", a numeric vector. If family = "binomial", a numeric/integer/logical vector with values in {0,1}. If family = "survival", a list with components time and status (1 = event, 0 = censored).
X	Design matrix of predictors. Can be a base matrix or something as.matrix() can coerce. No missing values are allowed.
family	Model family; one of c("normal","binomial","survival"). Determines which engine is called (S3VS_LM, S3VS_GLM, or S3VS_SURV).
cor_xy	Optional numeric vector of precomputed marginal correlations between y and each column of X. Used only when family="normal" to speed up or reproduce screening by $\|cor(y, X_j)\|$. If NULL, correlations are computed internally.

surv_model — Character string specifying the survival model (for family="survival" only). Must be explicitly provided; there is no default. Values are "Cox" for proportional hazards models, "AFT" for accelerated failure time models.

method_xy — Rule for screening some predictors as "leading variables" based on their association with the response; one of c("topk", "fixedthresh", "percthresh"). The association measure depends on family (e.g., correlation for "normal", eta-squared for "binomial", or marginal utility for "survival").

"topk" keeps the predictors with the largest k association values; "fixedthresh" keeps predictors whose association is greater than or equal to a specified threshold; "percthresh" keeps predictors whose association is within a given percentage of the best.

param_xy — Tuning parameter for method_xy. If "topk", supply a list with an integer k (keep the top k). If "fixedthresh", supply a list with a numeric threshold thresh (keep predictors with association $\geq$ threshold). If "percthresh", supply a list with a numeric percentage thresh in $(0, 100]$ (keep predictors with association $\geq$ that percent of the highest association).

method_xx — Rule for constructing, for each *leading variable*, the set of associated predictors (the "leading set") using inter-predictor association (absolute value of the correlation coefficient); one of c("topk", "fixedthresh", "percthresh") with same interpretation as method_xy.

param_xx — Tuning parameter for method_xx; same interpretation as param_xy but applied to inter-predictor association (absolute value of the correlation coefficient).

vsel_method — Character string specifying the variable selection method to be used within each *leading set*. Available options depend on the model type:

- For linear models (S3VS_LM) and generalized linear models (S3VS_GLM): "NLP", "LASSO", "ENET", "SCAD", "MCP".
- For survival models (S3VS_SURV): "LASSO", "ENET" for surv_model=COX and "AFTGEE", "BRIDGE", "PVAFT" for surv_model=AFT.

alpha — Only used when vsel_method == "ENET". Elastic net mixing parameter, with $\alpha \in (0, 1)$.

method_sel — Policy for aggregating predictors selected across leading sets in an iteration; one of c("conservative","liberal"). "conservative" selects the smallest admissible set of predictors by intersecting the selected sets of predictors across leading sets, beginning with all and gradually reducing from the end until a non-empty intersection is found; this ensures only predictors consistently selected across leading sets are retained. "liberal" selects the largest admissible set of predictors by taking the union of all selected sets of predictors, so any predictor chosen in at least one leading set is included. If no predictor is selected from the first leading set, the iteration does not contribute to final selection and exclusion rules (method_rem) are applied instead.

method_rem — Policy for excluding predictors when no selections are made in an iteration; one of c("conservative_begin","conservative_end","liberal"). "conservative_begin" excludes the smallest admissible set of predictors by intersecting the non-selected sets of predictors starting from the first leading set; "conservative_end" does the same but begins from the last leading set and moves backward; "liberal"

excludes the largest admissible set of predictors by taking the union of all non-selected sets of predictor. Predictors excluded under this rule are removed from subsequent iterations.

`sel_regout` Logical (GLM only). If `TRUE`, when predictors are selected in an iteration, the working response `y` is updated using the selected predictors via `update_y_GLM`. Ignored for other families.

`rem_regout` Logical (for LM and GLM only). If `TRUE`, when no predictors are selected in an iteration and some are removed instead, the working response `y` is updated using the removed predictors via `update_y_LM` or `update_y_GLM`. Ignored for other families.

`update_y_thresh`
Numeric scalar threshold controlling how the working response `y` is updated in GLMs when `sel_regout=TRUE` or `rem_regout=TRUE`. When $|y - fitted\_y| > update\_y\_thresh$, `y` is kept, else `y` replaced by the rounded value of fitted_y, where fitted_y is the fitted probability from the logistic model. The default value is 0.5. Ignored for other families.

`m` Integer. Maximum number of S3VS iterations to perform. Defaults to `100`.

`nskip` Integer. Maximum number of iterations in which no new predictors are selected before the algorithm stops. Defaults to `3`.

`verbose` Logical. If `TRUE`, prints detailed progress information at each iteration (e.g., iteration number, predictors selected or removed). Defaults to `FALSE`.

`seed` If supplied, sets the random seed via `set.seed()` to ensure reproducibility of stochastic components. If `NULL`, no seed is set.

`parallel` Logical. If `TRUE`, attempts to perform some computations in parallel mode in `binomial` and `survival` families, which is strongly recommended for faster execution. Defaults to `FALSE`.

Details

Model: For a continuous response, S3VS considers the linear model (LM)

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

For a binary response, S3VS considers the generalized linear model (GLM)

$$g(E(\boldsymbol{y} \mid \boldsymbol{X})) = \boldsymbol{X}\boldsymbol{\beta}$$

For a survival type response, S3VS considers two choices of models–the Cox model

$$\lambda(t \mid \boldsymbol{x}_i) = \lambda_0(t)\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})$$

and the AFT model

$$\log(\boldsymbol{T}) = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

S3VS algorithm: The general form of the S3VS algorithm consists of the following steps, repeated iteratively until convergence:

1. **Determination of leading variables:** 'Leading variables' are determined based on the association of the predictors with the response, following one of three rules. The rule is fixed by the arguments `method_xy` and `param_xy`.
2. **Determination of leading sets:** For each leading variable, a group of related predictors, called the 'leading set', is determined based on the association of all candidate predictors with the leading variable, following one of three rules. The rule is fixed by the arguments `method_xx` and `param_xx`.
3. **Variable selection:** Within each leading set, small to moderate-dimensional variable selection is performed using a method fixed by `vsel_method`.
4. **Aggregation of selected/not-selected variables:** Variables selected/not-selected in different leading sets are aggregated using several possible rules, fixed by `method_sel` and `method_rem`.
5. **Updation of response and/or set of covariates:** At the end of each iteration, the response and predictors may be chosen to be updated or not through argumentsm `sel_regout`, `rem_regout`, and `update_y_thresh`.

The convergence criterion is determined by the arguments `m` and `nkip` jointly. For ore details of the individual steps, see the manual of the functions linked below.

Value

A list with the following components:

`selected` A character vector of predictor names that were selected across all iterations.

`selected_iterwise` A list recording the predictors selected at each iteration, in the order they were considered.

`runtime` Runtime in seconds.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

See Also

`get_leadvars`, `get_leadsets`, `VS_method`, `select_vars`, `remove_vars`, `update_y`

Examples

```
### [1] For linear model
# Simulate continuous data
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
y <- X[,1] + 0.5 * X[,2] + rnorm(n)
# Run S3VS for LM
res_lm <- S3VS(y = y, X = X, family = "normal",
               method_xy = "topk", param_xy = list(k=1),
               method_xx = "topk", param_xx = list(k=3),
```

```
                vsel_method = "LASSO", method_sel = "conservative",
                method_rem = "conservative_begin", rem_regout = FALSE,
                m = 100, nskip = 3, verbose = TRUE, seed = 123)
# View selected predictors
res_lm$selected

### [2] For generalized linear model
# Simulate binary data
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
eta <- X[,1] + 0.5 * X[,2]
prob <- 1 / (1 + exp(-eta))
y <- rbinom(n, size = 1, prob = prob)
# Run S3VS for for GLM (logistic)
res_glm <- S3VS(y = y, X = X, family = "binomial",
                method_xy = "topk", param_xy = list(k = 1),
                method_xx = "topk", param_xx = list(k = 3),
                vsel_method = "LASSO",
                method_sel = "conservative", method_rem = "conservative_begin",
                sel_regout = FALSE, rem_regout = FALSE,
                m = 100, nskip = 3, verbose = TRUE, seed = 123)
# View selected predictors
res_glm$selected

### [3] For survival model
# Simulate survival data (Cox)
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
eta <- X[,1] + 0.5 * X[,2]
base_rate <- 0.05
T_event <- rexp(n, rate = base_rate * exp(eta))
C <- rexp(n, rate = 0.03)
time <- pmin(T_event, C)
status <- as.integer(T_event <= C)
y_surv <- list(time = time, status = status)
# Run S3VS for linear models
res_surv <- S3VS(y = y_surv, X = X, family = "survival",
                 surv_model = "COX",
                 method_xy = "topk", param_xy = list(k = 1),
                 method_xx = "topk", param_xx = list(k = 3),
                 vsel_method = "COXGLMNET",
                 method_sel = "conservative", method_rem = "conservative_begin",
                 sel_regout = FALSE, rem_regout = FALSE,
                 m = 100, nskip = 3, verbose = TRUE, seed = 123)
# View selected predictors
res_surv$selected
```

S3VS_GLM *Structured Screen-and-Select Variable Selection in Generalized Linear Models*

Description

S3VS_GLM performs variable selection based on the structured screen-and-select framework in generalized linear models.

Usage

```
S3VS_GLM(y, X,
  method_xy = c("topk", "fixedetasqthresh", "percetasqthresh"), param_xy,
  method_xx = c("topk", "fixedcorthresh", "perccorthresh"), param_xx,
  vsel_method = c("NLP", "LASSO", "ENET", "SCAD", "MCP"),
  alpha = 0.5,
  method_sel = c("conservative", "liberal"),
  method_rem = c("conservative_begin", "conservative_end", "liberal"),
  sel_regout = FALSE, rem_regout = FALSE, update_y_thresh = NULL,
  m = 100, nskip = 3, verbose = FALSE, seed = NULL, parallel = FALSE)
```

Arguments

y	Response. A numeric/integer/logical vector with values in {0,1}.
X	Design matrix of predictors. Can be a base matrix or something as.matrix() can coerce. No missing values are allowed.
method_xy	Rule for screening some predictors as "leading variables" based on their association with the response; one of c("topk", "fixedthresh", "percthresh"). The association measure is eta-squared. "topk" keeps the predictors with the largest k association values; "fixedthresh" keeps predictors whose association is greater than or equal to a specified threshold; "percthresh" keeps predictors whose association is within a given percentage of the best.
param_xy	Tuning parameter for method_xy. If "topk", supply a list with an integer k (keep the top k). If "fixedthresh", supply a list with a numeric threshold thresh (keep predictors with association $\geq$ threshold). If "percthresh", supply a list with a numeric percentage thresh in $(0, 100]$ (keep predictors with association $\geq$ that percent of the highest association).
method_xx	Rule for constructing, for each *leading variable*, the set of associated predictors (the "leading set") using inter-predictor association (absolute value of the correlation coefficient); one of c("topk", "fixedthresh", "percthresh") with same interpretation as method_xy.
param_xx	Tuning parameter for method_xx; same interpretation as param_xy but applied to inter-predictor association (absolute value of the correlation coefficient).
vsel_method	Character string specifying the variable selection method to be used within each *leading set*. Available options are "NLP", "LASSO", "ENET", "SCAD", "MCP".

`alpha` Only used when `vsel_method == "ENET"`. Elastic net mixing parameter, with $\alpha \in (0, 1)$.

`method_sel` Policy for aggregating predictors selected across leading sets in an iteration; one of `c("conservative","liberal")`. `"conservative"` selects the smallest admissible set of predictors by intersecting the selected sets of predictors across leading sets, beginning with all and gradually reducing from the end until a non-empty intersection is found; this ensures only predictors consistently selected across leading sets are retained. `"liberal"` selects the largest admissible set of predictors by taking the union of all selected sets of predictors, so any predictor chosen in at least one leading set is included. If no predictor is selected from the first leading set, the iteration does not contribute to final selection and exclusion rules (`method_rem`) are applied instead.

`method_rem` Policy for excluding predictors when no selections are made in an iteration; one of `c("conservative_begin","conservative_end","liberal")`. `"conservative_begin"` excludes the smallest admissible set of predictors by intersecting the non-selected sets of predictors starting from the first leading set; `"conservative_end"` does the same but begins from the last leading set and moves backward; `"liberal"` excludes the largest admissible set of predictors by taking the union of all non-selected sets of predictor. Predictors excluded under this rule are removed from subsequent iterations.

`sel_regout` Logical. If `TRUE`, when predictors are selected in an iteration, the working response `y` is updated using the selected predictors via `update_y_GLM`.

`rem_regout` Logical. If `TRUE`, when no predictors are selected in an iteration and some are removed instead, the working response `y` is updated using the removed predictors via `update_y_GLM`.

`update_y_thresh` Numeric scalar threshold controlling how the working response `y` is updated when `sel_regout=TRUE` or `rem_regout=TRUE`. When $|y - fitted\_y| > update\_y\_thresh$, `y` is kept, else `y` replaced by the rounded value of fitted_y, where fitted_y is the fitted probability from the logistic model. The default value is 0.5.

`m` Integer. Maximum number of S3VS iterations to perform. Defaults to `100`.

`nskip` Integer. Maximum number of iterations in which no new predictors are selected before the algorithm stops. Defaults to `3`.

`verbose` Logical. If `TRUE`, prints detailed progress information at each iteration (e.g., iteration number, predictors selected or removed). Defaults to `FALSE`.

`seed` If supplied, sets the random seed via `set.seed()` to ensure reproducibility of stochastic components. If `NULL`, no seed is set.

`parallel` Logical. If `TRUE`, attempts to perform some computations in parallel mode, which is strongly recommended for faster execution. Defaults to `FALSE`.

Details

For a binary response, S3VS considers the generalized linear model (GLM)

$$g(E(\boldsymbol{y} \mid \boldsymbol{X})) = \boldsymbol{X}\boldsymbol{\beta}$$

For the S3VS algorithm, see the manual of the top-level function `S3VS`.

Value

A list with the following components:

`selected`	A character vector of predictor names that were selected across all iterations.
`selected_iterwise`	A list recording the predictors selected at each iteration, in the order they were considered.
`runtime`	Runtime in seconds.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

See Also

`get_leadvars_GLM`, `get_leadsets`, `VS_method_GLM`, `select_vars`, `remove_vars`, `update_y_GLM`

Examples

```
# Simulate binary data
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
eta <- X[,1] + 0.5 * X[,2]
prob <- 1 / (1 + exp(-eta))
y <- rbinom(n, size = 1, prob = prob)
# Run S3VS for for GLM (logistic)
res_glm <- S3VS_GLM(y = y, X = X,
                method_xy = "topk", param_xy = list(k = 1),
                method_xx = "topk", param_xx = list(k = 3),
                vsel_method = "LASSO",
                method_sel = "conservative", method_rem = "conservative_begin",
                sel_regout = FALSE, rem_regout = FALSE,
                m = 100, nskip = 3, verbose = TRUE, seed = 123)
# View selected predictors
res_glm$selected
```

`S3VS_LM` *Structured Screen-and-Select Variable Selection in Linear Models*

Description

`S3VS_LM` performs variable selection based on the structured screen-and-select framework in linear models.

Usage

```
S3VS_LM(y, X, cor_xy = NULL,
  method_xy = c("topk", "fixedcorthresh", "perccorthresh"), param_xy,
  method_xx = c("topk", "fixedcorthresh", "perccorthresh"), param_xx,
  vsel_method = c("NLP", "LASSO", "ENET", "SCAD", "MCP"),
  alpha = 0.5,
  method_sel = c("conservative", "liberal"),
  method_rem = c("conservative_begin", "conservative_end", "liberal"),
  rem_regout = FALSE,
  m = 100, nskip = 3, verbose = FALSE, seed = NULL)
```

Arguments

y	Response. A numeric vector.
X	Design matrix of predictors. Can be a base matrix or something as.matrix() can coerce. No missing values are allowed.
cor_xy	Optional numeric vector of precomputed marginal correlations between y and each column of X. Used to speed up or reproduce screening by $\|cor(y, X_j)\|$. If NULL, correlations are computed internally.
method_xy	Rule for screening some predictors as 'leading variables' based on their association with the response; one of c("topk", "fixedthresh", "percthresh"). The association measure is correlation. "topk" keeps the predictors with the largest k association values; "fixedthresh" keeps predictors whose association is greater than or equal to a specified threshold; "percthresh" keeps predictors whose association is within a given percentage of the best.
param_xy	Tuning parameter for method_xy. If "topk", supply a list with an integer k (keep the top k). If "fixedthresh", supply a list with a numeric threshold thresh (keep predictors with association $\geq$ threshold). If "percthresh", supply a list with a numeric percentage thresh in $(0, 100]$ (keep predictors with association $\geq$ that percent of the highest association).
method_xx	Rule for constructing, for each *leading variable*, the set of associated predictors (the "leading set") using inter-predictor association (absolute value of the correlation coefficient); one of c("topk", "fixedthresh", "percthresh") with same interpretation as method_xy.
param_xx	Tuning parameter for method_xx; same interpretation as param_xy but applied to inter-predictor association (absolute value of the correlation coefficient).
vsel_method	Character string specifying the variable selection method to be used within each *leading set*. Available options are "NLP", "LASSO", "ENET", "SCAD", "MCP".
alpha	Only used when vsel_method == "ENET". Elastic net mixing parameter, with $\alpha \in (0, 1)$.
method_sel	Policy for aggregating predictors selected across leading sets in an iteration; one of c("conservative","liberal"). "conservative" selects the smallest admissible set of predictors by intersecting the selected sets of predictors across leading sets, beginning with all and gradually reducing from the end until a non-empty intersection is found; this ensures only predictors consistently selected

across leading sets are retained. "liberal" selects the largest admissible set of predictors by taking the union of all selected sets of predictors, so any predictor chosen in at least one leading set is included. If no predictor is selected from the first leading set, the iteration does not contribute to final selection and exclusion rules (method_rem) are applied instead.

method_rem — Policy for excluding predictors when no selections are made in an iteration; one of c("conservative_begin","conservative_end","liberal"). "conservative_begin" excludes the smallest admissible set of predictors by intersecting the non-selected sets of predictors starting from the first leading set; "conservative_end" does the same but begins from the last leading set and moves backward; "liberal" excludes the largest admissible set of predictors by taking the union of all non-selected sets of predictor. Predictors excluded under this rule are removed from subsequent iterations.

rem_regout — Logical. If TRUE, when no predictors are selected in an iteration and some are removed instead, the working response y is updated using the removed predictors via update_y_LM.

m — Integer. Maximum number of S3VS iterations to perform. Defaults to 100.

nskip — Integer. Maximum number of iterations in which no new predictors are selected before the algorithm stops. Defaults to 3.

verbose — Logical. If TRUE, prints detailed progress information at each iteration (e.g., iteration number, predictors selected or removed). Defaults to FALSE.

seed — If supplied, sets the random seed via set.seed() to ensure reproducibility of stochastic components. If NULL, no seed is set.

Details

For a continuous response, S3VS considers the linear model (LM)

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

For the S3VS algorithm, see the manual of the top-level function S3VS.

Value

A list with the following components:

selected — A character vector of predictor names that were selected across all iterations.

selected_iterwise — A list recording the predictors selected at each iteration, in the order they were considered.

runtime — Runtime in seconds.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

See Also

get_leadvars_LM, get_leadsets, VS_method_LM, select_vars, remove_vars, update_y_LM

Examples

```
# Simulate continuous data
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
y <- X[,1] + 0.5 * X[,2] + rnorm(n)
# Run S3VS for LM
res_lm <- S3VS_LM(y = y, X = X,
                method_xy = "topk", param_xy = list(k=1),
                method_xx = "topk", param_xx = list(k=3),
                vsel_method = "LASSO", method_sel = "conservative",
                method_rem = "conservative_begin", rem_regout = FALSE,
                m = 100, nskip = 3, verbose = TRUE, seed = 123)
# View selected predictor
res_lm$selected
```

S3VS_SURV *Structured Screen-and-Select Variable Selection in Survival Models*

Description

S3VS_SURV performs variable selection based on the structured screen-and-select framework in survival models.

Usage

```
S3VS_SURV(y, X, surv_model = c("COX", "AFT"),
  method_xy = c("topk", "fixedmuthresh", "percmuthresh"), param_xy,
  method_xx = c("topk", "fixedcorthresh", "perccorthresh"), param_xx,
  vsel_method = c("LASSO", "ENET", "AFTGEE", "BRIDGE", "PVAFT"),
  alpha = 0.5,
  method_sel = c("conservative", "liberal"),
  method_rem = c("conservative_begin", "conservative_end", "liberal"),
  m = 100, nskip = 3, verbose = FALSE, seed = NULL, parallel = FALSE)
```

Arguments

y	Response. A list with components time and status (1 = event, 0 = censored).
X	Design matrix of predictors. Can be a base matrix or something as.matrix() can coerce. No missing values are allowed.
surv_model	Character string specifying the survival model. Must be explicitly provided; there is no default. Values are "Cox" for proportional hazards models, "AFT" for accelerated failure time models.

`method_xy` Rule for screening some predictors as "leading variables" based on their association with the response; one of `c("topk", "fixedthresh", "percthresh")`. The association measure is marginal utility.

`"topk"` keeps the predictors with the largest k association values; `"fixedthresh"` keeps predictors whose association is greater than or equal to a specified threshold; `"percthresh"` keeps predictors whose association is within a given percentage of the best.

`param_xy` Tuning parameter for `method_xy`. If `"topk"`, supply a list with an integer `k` (keep the top k). If `"fixedthresh"`, supply a list with a numeric threshold `thresh` (keep predictors with association $\geq$ threshold). If `"percthresh"`, supply a list with a numeric percentage `thresh` in $(0, 100]$ (keep predictors with association $\geq$ that percent of the highest association).

`method_xx` Rule for constructing, for each *leading variable*, the set of associated predictors (the "leading set") using inter-predictor association (absolute value of the correlation coefficient); one of `c("topk", "fixedthresh", "percthresh")` with same interpretation as `method_xy`.

`param_xx` Tuning parameter for `method_xx`; same interpretation as `param_xy` but applied to inter-predictor association (absolute value of the correlation coefficient).

`vsel_method` Character string specifying the variable selection method to be used within each *leading set*. Available options are `"LASSO"`, `"ENET"` for `surv_model=COX` and `"AFTGEE"`, `"BRIDGE"`, `"PVAFT"` for `surv_model=AFT`.

`alpha` Only used when `vsel_method == "ENET"`. Elastic net mixing parameter, with $\alpha \in (0, 1)$.

`method_sel` Policy for aggregating predictors selected across leading sets in an iteration; one of `c("conservative","liberal")`. `"conservative"` selects the smallest admissible set of predictors by intersecting the selected sets of predictors across leading sets, beginning with all and gradually reducing from the end until a non-empty intersection is found; this ensures only predictors consistently selected across leading sets are retained. `"liberal"` selects the largest admissible set of predictors by taking the union of all selected sets of predictors, so any predictor chosen in at least one leading set is included. If no predictor is selected from the first leading set, the iteration does not contribute to final selection and exclusion rules (`method_rem`) are applied instead.

`method_rem` Policy for excluding predictors when no selections are made in an iteration; one of `c("conservative_begin","conservative_end","liberal")`. `"conservative_begin"` excludes the smallest admissible set of predictors by intersecting the non-selected sets of predictors starting from the first leading set; `"conservative_end"` does the same but begins from the last leading set and moves backward; `"liberal"` excludes the largest admissible set of predictors by taking the union of all non-selected sets of predictor. Predictors excluded under this rule are removed from subsequent iterations.

`m` Integer. Maximum number of S3VS iterations to perform. Defaults to `100`.

`nskip` Integer. Maximum number of iterations in which no new predictors are selected before the algorithm stops. Defaults to `3`.

`verbose` Logical. If `TRUE`, prints detailed progress information at each iteration (e.g., iteration number, predictors selected or removed). Defaults to `FALSE`.

seed	If supplied, sets the random seed via `set.seed()` to ensure reproducibility of stochastic components. If `NULL`, no seed is set.
parallel	Logical. If `TRUE`, attempts to perform some computations in parallel mode, which is strongly recommended for faster execution. Defaults to `FALSE`.

Details

For a survival type response, S3VS considers two choices of models–the Cox model

$$\lambda(t \mid \boldsymbol{x}_i) = \lambda_0(t) \exp(\boldsymbol{x}_i^T \boldsymbol{\beta})$$

and the AFT model

$$\log(\boldsymbol{T}) = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

For the S3VS algorithm, see the manual of the top-level function `S3VS`.

Value

A list with the following components:

selected	A character vector of predictor names that were selected across all iterations.
selected_iterwise	A list recording the predictors selected at each iteration, in the order they were considered.
runtime	Runtime in seconds.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

See Also

`get_leadvars_SURV`, `get_leadsets`, `VS_method_SURV`, `select_vars`, `remove_vars`

Examples

```
# Simulate survival data (Cox)
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
eta <- X[,1] + 0.5 * X[,2]
base_rate <- 0.05
T_event <- rexp(n, rate = base_rate * exp(eta))
C <- rexp(n, rate = 0.03)
time <- pmin(T_event, C)
status <- as.integer(T_event <= C)
y_surv <- list(time = time, status = status)
# Run S3VS for linear models
res_surv <- S3VS(y = y_surv, X = X, family = "survival",
```

```
                  surv_model = "COX",
                  method_xy = "topk", param_xy = list(k = 1),
                  method_xx = "topk", param_xx = list(k = 3),
                  vsel_method = "COXGLMNET",
                  method_sel = "conservative", method_rem = "conservative_begin",
                  sel_regout = FALSE, rem_regout = FALSE,
                  m = 100, nskip = 3, verbose = TRUE, seed = 123)
# View selected predictors
res_surv$selected
```

`select_vars` *Aggregate Selected Predictors Across Multiple Leading Sets*

Description

`select_vars` combines variable selections obtained from multiple leading sets into a single set, using either a liberal (union) or conservative (progressive intersection) rule.

Usage

```
select_vars(listselect, method = c("conservative", "liberal"))
```

Arguments

`listselect`	A `list` of vectors, each containing names of the predictors selected in the corresponding leading set.
`method`	Aggregation rule. One of `"conservative"` or `"liberal"` (partial matching allowed). Referring to the sets (vectors included in `listselect`) of selected predictors:
	`"liberal"` Returns the *union* of all sets: `unique(unlist(listselect))`.
	`"conservative"` Returns the *last non-empty intersection* when intersecting the first $i = 1, 2, \ldots$ sets in order. The procedure stops once the running intersection becomes empty and returns the previous (last non-empty) intersection. If the first set is empty, returns `character(0)`. Order of vectors in `listselect` matters for this method.

Details

The liberal rule favors inclusiveness, while the conservative rule favors stability.

Value

Vector with names of the retained predictors (considered selected in the current iteration of S3VS); if no predictors are retained, `character(0)`).

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

Examples

```
listselect <- list(
  c("V1","V2","V23"),
  c("V4","V2","V23"),
  c("V4","V5","V23")
)
select_vars(listselect, method="conservative")
```

---

update_y *Update Response Accounting for Selected Predictors*

Description

update_y updates the response accounting for the selected predictors in linear models, and selected or removed predictors in generalized linear models.

Usage

```
update_y(y, X, family, vars, update_y_thresh = NULL)
```

Arguments

y	Response. If family = "normal", a numeric vector. If family = "binomial", a numeric/integer/logical vector with values in {0,1}.
family	Model family; one of c("normal","binomial"). Determines which engine is called (update_y_LM, update_y_GLM).
X	Predictor matrix. Can be a base matrix or something as.matrix() can coerce. No missing values are allowed.
vars	Character vector containing the names of predictors that need to be accounted for. They must appear in X.
update_y_thresh	Numeric scalar threshold used only family = "binomial". When $\|y - fitted\_y\| > update\_y\_thresh$, y is kept, else y replaced by the rounded value of $fitted_y$, which is the fitted probability from the logistic model. The default value is 0.5.

Value

Returns the updated response vector.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

See Also

update_y_LM, update_y_GLM

Examples

```
# Simulate continuous data
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
y <- X[,1] + 0.5 * X[,2] + rnorm(n)
update_y(y = y, X = X, family = "normal", vars = c("V1","V4"))
```

`update_y_GLM` *Update Response Accounting for Selected Predictors in Generalized Linear Models*

Description

`update_y_LM` updates the response accounting for the selected predictors in generalized linear models.

Usage

```
update_y_GLM(y, X, vars, update_y_thresh)
```

Arguments

`y`	Response. A numeric/integer/logical vector with values in {0,1}.
`X`	Predictor matrix. Can be a base matrix or something `as.matrix()` can coerce. No missing values are allowed.
`vars`	Character vector containing the names of predictors that need to be accounted for. They must appear in `X`.
`update_y_thresh`	Numeric scalar threshold. When $\|y - fitted\_y\| > update\_y\_thresh$, `y` is kept, else `y` replaced by the rounded value of fitted_y, which is the fitted probability from the logistic model. The default value is 0.5.

Value

Returns the updated (binary) response vector.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

Examples

```
# Simulate binary data
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
eta <- X[,1] + 0.5 * X[,2]
prob <- 1 / (1 + exp(-eta))
y <- rbinom(n, size = 1, prob = prob)
update_y(family = "binomial", y = y, X = X, vars = c("V1","V4"), update_y_thresh = 0.8)
```

---

update_y_LM *Update Response Accounting for Selected Predictors in Linear Models*

Description

`update_y_LM` updates the response accounting for the selected predictors in linear models.

Usage

```
update_y_LM(y, X, vars)
```

Arguments

`y`	Response. A numeric vector of length n.
`X`	Predictor matrix. Can be a base matrix or something `as.matrix()` can coerce. No missing values are allowed.
`vars`	Character vector containing the names of predictors that need to be accounted for. They must appear in `X`.

Value

Returns the updated response vector.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

Examples

```
# Simulate continuous data
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
y <- X[,1] + 0.5 * X[,2] + rnorm(n)
update_y(family = "normal", y = y, X = X, vars = c("V1","V4"))
```

VS_method *Variable Selection in Leading Sets under the S3VS Framework*

Description

VS_method applies the chosen variable-selection algorithm to each leading set produced by S3VS at every iteration.

Usage

```
VS_method(y, X, family, surv_model = NULL, vsel_method, alpha = 0.5,
          p_thresh = 0.1, gamma = 0.9, verbose = FALSE)
```

Arguments

y	Response. If family = "normal", a numeric vector. If family = "binomial", a numeric/integer/logical vector with values in {0,1}. If family = "survival", a list with components time and status (1 = event, 0 = censored).
X	Predictor matrix. Can be a base matrix or something as.matrix() can coerce. No missing values are allowed.
family	Model family; one of c("normal","binomial","survival"). Determines which engine is called (VS_method_LM, VS_method_GLM, or VS_method_SURV).
surv_model	Character string specifying the survival model (family="survival" only). Must be explicitly provided; there is no default. Values are "Cox" for proportional hazards models, "AFT" for accelerated failure time models.
vsel_method	Character string indicating the variable-selection engine used inside S3VS at each iteration. Allowed values depend on family: for family = "normal"(S3VS_LM) and "binomial" (S3VS_GLM), available options are "NLP", "ENET", "LASSO", "SCAD", "MCP"; for family = "survival", available options are "LASSO", "ENET" for surv_model = "COX" and "AFTGEE", "BRIDGE", "PVAFT" for surv_model = "AFT". See Details for more information.
alpha	Only used when vsel_method == "ENET". Elastic net mixing parameter, with $\alpha \in (0, 1)$.
p_thresh	Only used for surv_model = "AFT" with vsel_method = "AFTGEE". p-value threshold for variable selection.
gamma	Only used for surv_model = "AFT" with vsel_method = "BRIDGE". Numeric scalar (default 0.5) giving the exponent in the bridge penalty. Must be > 0. See details.
verbose	If TRUE, some information is printed. FALSE, by default.

Details

Details to come...

Value

A list containing:

sel	Character vector with names of the selected predictors.
nosel	Character vector with names of the predictors not selected.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

See Also

VS_method_LM, VS_method_GLM, VS_method_SURV

Examples

```
# Simulate continuous data
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
y <- X[,1] + 0.5 * X[,2] + rnorm(n)
# Run VS_method
VS_method(y, X, family = "normal", vsel_method = "NLP", verbose = FALSE)
```

VS_method_GLM *Variable Selection in Leading Sets for Generalized Linear Models under the S3VS Framework*

Description

VS_method applies the chosen variable-selection algorithm for generalized linear models to each leading set produced by S3VS at every iteration.

Usage

```
VS_method_GLM(y, X, vsel_method, alpha = 0.5, verbose = FALSE,
              parallel = FALSE, ncores = NULL)
```

Arguments

y	Response. A numeric/integer/logical vector with values in {0,1}.
X	Predictor matrix. Can be a base matrix or something as.matrix() can coerce. No missing values are allowed.
vsel_method	Character string indicating the variable-selection engine used at each iteration. Available options are "NLP" and "LASSO". See Details for more information.

alpha	Only used when vsel_method == "ENET". Elastic net mixing parameter, with $\alpha \in (0, 1)$.
verbose	If TRUE, some information is printed. FALSE, by default.
parallel	Logical. If TRUE, cross-validation steps in penalized regression methods (e.g., LASSO or elastic net via glmnet) are executed in parallel using the **doParallel** backend. Defaults to FALSE. The package **doParallel** must be installed when parallel = TRUE.
ncores	Integer; number of CPU cores to use when parallel = TRUE. If NULL (default), the number of cores is set to one fewer than the total number detected on the system. Ignored when parallel = FALSE.

Value

A list containing:

sel	Character vector with names of the selected predictors.
nosel	Character vector with names of the predictors not selected.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

See Also

modelSelection, cv.glmnet, cv.ncvreg

Examples

```
# Simulate binary data
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
eta <- X[,1] + 0.5 * X[,2]
prob <- 1 / (1 + exp(-eta))
y <- rbinom(n, size = 1, prob = prob)
# Run VS_method
VS_method_GLM(y, X, vsel_method = "LASSO", verbose = FALSE)
```

VS_method_LM *Variable Selection in Leading Sets for Linear Models under the S3VS Framework*

Description

VS_method applies the chosen variable-selection algorithm for linear models to each leading set produced by S3VS at every iteration.

Usage

```
VS_method_LM(y, X, vsel_method, alpha = 0.5, verbose = FALSE)
```

Arguments

y	Response. A numeric vector.
X	Predictor matrix. Can be a base matrix or something `as.matrix()` can coerce. No missing values are allowed.
vsel_method	Character string indicating the variable-selection engine used at each iteration. Available options are `"NLP"`, `"LASSO"`, `"SCAD"`, `"MCP"`. See Details for more information.
alpha	Only used when `vsel_method == "ENET"`. Elastic net mixing parameter, with $\alpha \in (0, 1)$.
verbose	If `TRUE`, some information is printed. `FALSE`, by default.

Value

A list containing:

sel	Character vector with names of the selected predictors.
nosel	Character vector with names of the predictors not selected.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

See Also

modelSelection, cv.glmnet, cv.ncvreg

Examples

```
# Simulate continuous data
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
y <- X[,1] + 0.5 * X[,2] + rnorm(n)
# Run VS_method
VS_method_LM(y, X, vsel_method = "NLP", verbose = FALSE)
```

VS_method_SURV *Variable Selection in Leading Sets for Survival Models under the S3VS Framework*

Description

VS_method applies the chosen variable-selection algorithm for survival models to each leading set produced by S3VS at every iteration.

Usage

```
VS_method_SURV(y, X, surv_model, vsel_method, alpha = 0.5,
  p_thresh = 0.1, gamma = 0.9, verbose = FALSE, ...)
```

Arguments

y	Response. A list with components time and status (1 = event, 0 = censored).
X	Predictor matrix. Can be a base matrix or something as.matrix() can coerce. No missing values are allowed.
surv_model	Character string specifying the survival model. Must be explicitly provided; there is no default. Values are "Cox" for proportional hazards models, "AFT" for accelerated failure time models.
vsel_method	Character string indicating the variable-selection engine used at each iteration. Available options are "COXGLMNET" for surv_model = "COX" and "AFTREG", "AFTGEE", "BRIDGE", and "PVAFT" for surv_model = "AFT". See Details for more information.
alpha	Only used when vsel_method == "ENET". Elastic net mixing parameter, with $\alpha \in (0, 1)$.
p_thresh	Only used with vsel_method = "AFTGEE". p-value threshold for variable selection.
gamma	Only used with vsel_method = "BRIDGE". Numeric scalar (default 0.5) giving the exponent in the bridge penalty. Must be > 0. See details.
verbose	If TRUE, some information is printed. FALSE, by default.
...	Other arguments to be passed inside eha::aftreg() when surv_model = "AFT" and vsel_method = "AFTREG"

Value

A list containing:

sel	Character vector with names of the selected predictors.
nosel	Character vector with names of the predictors not selected.

Author(s)

Nilotpal Sanyal <nsanyal@utep.edu>, Padmore N. Prempeh <pprempeh@albany.edu>

See Also

cv.glmnet, aftreg, aftgee, bridge_aft, pvaft

Examples

```
# Simulate survival data (Cox)
set.seed(123)
n <- 100
p <- 150
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
eta <- X[,1] + 0.5 * X[,2]
base_rate <- 0.05
T_event <- rexp(n, rate = base_rate * exp(eta))
C <- rexp(n, rate = 0.03)
time <- pmin(T_event, C)
status <- as.integer(T_event <= C)
y_surv <- list(time = time, status = status)
# Run VS_method
VS_method_SURV(y_surv, X, surv_model = "COX", vsel_method = "COXGLMNET", verbose = FALSE)
```

Index