

Package: combss (via r-universe)

May 11, 2026

Type Package

Title Continuous Optimisation Towards Best Subset Selection

Version 0.1.0

Author Benoit Liquet [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-8136-2294>>), Anant Mathur [aut],
Sarat Moka [aut]

Maintainer Benoit Liquet <benoit.liquet@univ-pau.fr>

Description Best subset selection in generalised linear models via continuous optimisation. Reformulates the NP-hard discrete subset selection problem as a continuous optimisation over the hypercube $[0,1]^p$, solved via a Frank-Wolfe homotopy algorithm with closed-form ridge inner solves. Supports linear (Gaussian), binary logistic, and multinomial regression. For methodological details see Moka, Liquet, Zhu and Muller (2024) <[doi:10.1007/s11222-024-10387-8](https://doi.org/10.1007/s11222-024-10387-8)> and Mathur, Liquet, Muller and Moka (2026) <[doi:10.48550/arXiv.2603.21952](https://doi.org/10.48550/arXiv.2603.21952)>.

License GPL-3

URL <https://github.com/benoit-liquet/combss>

Encoding UTF-8

Imports glmnet (>= 4.0), stats

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository <https://cran.r-universe.dev>

Date/Publication 2026-05-11 21:45:43 UTC

RemoteUrl <https://github.com/cran/combss>

RemoteRef HEAD

RemoteSha 27b27725d35294f943e1ac131d70739c9beef1af

Contents

coef.combss	2
combss	3
combss_cv	4
predict.combss	5
Index	7

coef.combss	<i>Extract the subset of selected feature indices.</i>
-------------	--

Description

Extract the subset of selected feature indices.

Usage

```
## S3 method for class 'combss'
coef(object, k = NULL, ...)
```

Arguments

object	A combss object.
k	Subset size to return. Defaults to the best k if validation data was used, else the largest k evaluated.
...	Unused.

Value

Integer vector of selected column indices (1-indexed).

Examples

```
set.seed(1)
x <- matrix(rnorm(100 * 20), 100, 20)
y <- as.numeric(x[, 1] + x[, 2] + rnorm(100) * 0.1)
fit <- combss(x, y, family = "gaussian", q = 5)
coef(fit, k = 2)
```

combss	<i>Best subset selection for generalised linear models via continuous optimisation (COMBSS).</i>
--------	--

Description

Reformulates the NP-hard discrete subset selection problem as a continuous optimisation over the hypercube $[0, 1]^p$, solved via a Frank-Wolfe homotopy algorithm. The inner ridge problem is solved with `glmnet`. For each subset size $k = 1, \dots, q$, COMBSS returns the selected feature set; if validation data is supplied, the best k is chosen by validation MSE (Gaussian) or accuracy (binomial / multinomial).

Usage

```
combss(
  x,
  y,
  family = c("gaussian", "binomial", "multinomial"),
  q = NULL,
  x_val = NULL,
  y_val = NULL,
  lam_ridge = 0,
  Niter = 25,
  alpha = 0.01,
  scale = TRUE,
  mandatory = NULL,
  verbose = FALSE
)
```

Arguments

<code>x</code>	Numeric design matrix (n, p) without an intercept column.
<code>y</code>	Response. Numeric for <code>family = "gaussian"</code> ; $\{0, 1\}$ (or two-level factor) for <code>family = "binomial"</code> ; factor or integer in $\{1, \dots, C\}$ for <code>family = "multinomial"</code> .
<code>family</code>	One of "gaussian" (alias: "linear"), "binomial", "multinomial".
<code>q</code>	Maximum subset size. Defaults to $\min(n, p)$.
<code>x_val, y_val</code>	Optional validation data. When provided, the best k is selected by validation metric and <code>coef_</code> is returned for the best subset.
<code>lam_ridge</code>	Ridge penalty for the inner solver. Default 0.
<code>Niter</code>	Number of homotopy iterations. Default 25.
<code>alpha</code>	Frank-Wolfe step size. Default 0.01.
<code>scale</code>	Column-normalise <code>x</code> before the algorithm. Default TRUE.
<code>mandatory</code>	Optional integer vector of 1-indexed columns of <code>x</code> that must be included in every model.
<code>verbose</code>	Print progress. Default FALSE.

Value

An object of class "combss", a list with elements:

- family: one of "gaussian", "binomial", "multinomial"
- subset_list: list of length up to q; element k is an integer vector of selected column indices (1-indexed) for subset size k
- subset: best subset (requires validation data; otherwise NULL)
- mse / accuracy: validation metric for the best subset (requires validation data)
- lam_ridge: ridge penalty used
- q: maximum subset size evaluated

References

Moka, S., Liquet, B., Zhu, H. and Muller, S. (2024). COMBSS: best subset selection via continuous optimization. *Statistics and Computing*. doi:10.1007/s11222024103878

Mathur, A., Liquet, B., Muller, S. and Moka, S. (2026). Parsimonious Subset Selection for Generalized Linear Models with Biomedical Applications. [arXiv:2603.21952](https://arxiv.org/abs/2603.21952).

Examples

```
set.seed(1)
n <- 200; p <- 30
beta <- c(3, 2, 1.5, 1, 0.5, rep(0, p - 5))
x <- matrix(rnorm(n * p), n, p)
y <- as.numeric(x %*% beta + rnorm(n) * 0.5)
fit <- combss(x, y, family = "gaussian", q = 10)
fit$subset_list[1:5]
```

combss_cv

Select the ridge penalty for COMBSS by leave-one-out cross-validation.

Description

Mirrors `combss.cv.select_lambda` in the Python `combss` package. For each candidate `lambda` and each subset size `k = 1, \dots, q`, runs COMBSS to obtain a subset and evaluates LOOCV error of a refit. Returns the best `lambda` overall and the best `lambda` per `k`.

Usage

```
combss_cv(
  x,
  y,
  family = c("gaussian", "binomial", "multinomial"),
  q,
```

```

    lambda_grid = NULL,
    Niter = 25,
    verbose = FALSE
  )

```

Arguments

x	Numeric design matrix (n, p).
y	Response (numeric / binary / factor depending on family).
family	One of "gaussian" (alias "linear"), "binomial", "multinomial".
q	Maximum subset size.
lambda_grid	Candidate ridge penalties. Defaults to <code>c(0, 10^seq(-3, 1, length.out = 10))</code> .
Niter	Number of homotopy iterations passed to <code>combss()</code> .
verbose	Print progress.

Value

List with:

- `best_lambda`: ridge penalty with best mean LOOCV metric across k
- `best_lambda_per_k`: numeric vector, length q
- `results`: data frame (lambda, k, loocv_metric)

predict.combss	<i>Predict from a combss fit.</i>
----------------	-----------------------------------

Description

Refits the chosen subset on the original training data via OLS or ridge (Gaussian / binomial / multinomial as appropriate) and predicts on `newx`. Note: the training data is not stored on the fit object, so `newx` must be supplied along with `x_train` and `y_train`.

Usage

```

## S3 method for class 'combss'
predict(object, newx, x_train, y_train, k = NULL, ...)

```

Arguments

object	A combss object.
newx	Numeric matrix of new observations.
x_train, y_train	Original training data used to refit on the chosen subset.
k	Subset size to use. Defaults to the best k if validation data was used, else the largest k evaluated.
...	Unused.

Value

Numeric vector of predictions for newx.

Examples

```
set.seed(1)
x <- matrix(rnorm(100 * 20), 100, 20)
y <- as.numeric(x[, 1] + x[, 2] + rnorm(100) * 0.1)
fit <- combss(x, y, family = "gaussian", q = 5)
newx <- matrix(rnorm(5 * 20), 5, 20)
predict(fit, newx = newx, x_train = x, y_train = y, k = 2)
```

Index

`coef.combss`, 2

`combss`, 3

`combss_cv`, 4

`predict.combss`, 5