Package 'sparselink'

June 3, 2025

Version 1.0.0

Title Sparse Regression for Related Problems

Description Estimates sparse regression models (i.e., with few non-zero coefficients) in high-dimensional multi-
task learning and transfer learning settings, as proposed by Rauschenberger et al. (2025) <https://orbilu.uni.lu/handle/10993/63425>.

Depends R (>= 3.0.0)

Imports glmnet, pROC, stats, mvtnorm, spls, xrnet

Suggests knitr, testthat, remotes, glmtrans, rmarkdown

Encoding UTF-8

VignetteBuilder knitr

RoxygenNote 7.3.2

URL https://github.com/rauschenberger/sparselink, https://rauschenberger.github.io/sparselink/

BugReports https://github.com/rauschenberger/sparselink/issues

License MIT + file LICENSE

NeedsCompilation no

Author Armin Rauschenberger [aut, cre] (ORCID: <https://orcid.org/0000-0001-6498-4801>)

Maintainer Armin Rauschenberger <armin.rauschenberger@lih.lu>

Repository CRAN

Date/Publication 2025-06-03 09:30:05 UTC

Contents

sparselink-package *Sparse regression for related problems*

Description

The R package ‘sparselink‘ implements sparse regression for related problems (multi-task learning and transfer learning).

Details

Use function [sparselink()] for model fitting. Type ‘library(sparselink)‘ and then ‘?sparselink‘ or ‘help("sparselink")‘ to open its help file.

See the vignette for further examples. Type ‘vignette("sparselink")‘ or ‘browseVignettes("sparselink")‘ to open the vignette.

Author(s)

Maintainer: Armin Rauschenberger <armin.rauschenberger@lih.lu> (ORCID)

References

Armin Rauschenberger, Petr N. Nazarov, and Enrico Glaab (2025). "Estimating sparse regression models in multi-task learning and transfer learning through adaptive penalisation". *Under revision*. https://hdl.handle.net/10993/63425

See Also

First use `sparselink` to fit the models, and then `coef` to extract coefficients or `predict` to make predictions.

Examples

```
?sparselink
?coef.sparselink
?predict.sparselink
```

coef.sparselink *Regression Coefficients*

Description

Extracts coefficients from multi-task or transfer learning regression model.

Usage

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

Arguments

`object`	object of class `"sparselink"` (generated by function sparselink)
`...`	(not applicable)

Value

Returns estimated coefficients. The output is a list with two slots: slot `alpha` with the estimated intercept (vector of length q), and slot `beta` with the estimated slopes (matrix with p rows and q columns).

References

Armin Rauschenberger, Petr N. Nazarov, and Enrico Glaab (2025). "Estimating sparse regression models in multi-task learning and transfer learning through adaptive penalisation". *Under revision*. https://hdl.handle.net/10993/63425

See Also

Use `sparselink` to fit the model and `predict` to make predictions.

Examples

```
family <- "gaussian"
type <- "multiple" #  try "multiple" or "transfer"
if(type=="multiple"){
 data <- sim_data_multi(family=family)
} else if(type=="transfer"){
 data <- sim_data_trans(family=family)
}

object <- sparselink(x=data$X_train,y=data$y_train,family=family)
coef <- coef(object=object)
```

`predict.sparselink` *Out-of-sample Predictions*

Description

Predicts outcomes with a multi-task or transfer learning regression model.

Usage

```
## S3 method for class 'sparselink'
predict(object, newx, weight = NULL, ...)
```

Arguments

`object`	object of class `"sparselink"` (generated by function sparselink)
`newx`	features: matrix with n rows (samples) and p columns (variables) for multi-task learning; list of q matrices with n_k rows (samples) and p columns (variables) for transfer learning, for each k in $1, \ldots, q$
`weight`	hyperparameters for scaling external and internal weights: numeric vector of length 2, with the first entry for the external weights (prior coefficients from source data), and the second entry for the internal weights (prior coefficients from target data), selected values must be among the candidate values, default: `NULL` (using cross-validated weights)
`...`	(not applicable)

Value

Returns predicted values or predicted probabilities. The output is a list of q column vectors of length n_k for k in $1, \ldots, q$. Each vector corresponds to one target (multi-task learning) or one dataset (transfer learning).

References

Armin Rauschenberger, Petr N. Nazarov, and Enrico Glaab (2025). "Estimating sparse regression models in multi-task learning and transfer learning through adaptive penalisation". *Under revision*. https://hdl.handle.net/10993/63425

See Also

Use `sparselink` to fit the model and `coef` to extract coefficients.

Examples

```
family <- "gaussian"
type <- "multiple" # try "multiple" or "transfer"
if(type=="multiple"){
 data <- sim_data_multi(family=family)
} else if(type=="transfer"){
 data <- sim_data_trans(family=family)
}

object <- sparselink(x=data$X_train,y=data$y_train,family=family)
y_hat <- predict(object=object,newx=data$X_test)
```

`sparselink` *Sparse regression for related problems*

Description

Estimates sparse regression models (i.e., performing feature selection) in multi-task learning or transfer learning. Multi-task learning involves multiple targets, and transfer learning involves multiple datasets.

Usage

```
sparselink(
  x,
  y,
  family,
  alpha.init = 0.95,
  alpha = 1,
  type = "exp",
  nfolds = 10,
  cands = NULL
)
```

Arguments

`x`	$n \times p$ matrix (multi-task learning) or list of $n_k \times p$ matrices (transfer learning)
`y`	$n \times q$ matrix (multi-task learning) or list of n_k-dimensional vectors (transfer learning)
`family`	character `"gaussian"` or `"binomial"`
`alpha.init`	elastic net mixing parameter for initial regressions, default: 0.95 (lasso-like elastic net)
`alpha`	elastic net mixing parameter of final regressions, default: 1 (lasso)
`type`	default `"exp"` scales weights with $w_{ext}^{v_{ext}} + w_{int}^{v_{int}}$ (see internal function construct_penfacs for details)
`nfolds`	number of internal cross-validation folds, default: 10 (10-fold cross-validation)
`cands`	candidate values for both scaling parameters, default: `NULL` ({0, 0.2, 0.4, 0.6, 0.8, 1})

Value

Returns an object of class `sparselink`, a list with multiple slots:

- Stage 1 regressions (before sharing information): Slot `glm.one` contains q objects of type `cv.glmnet` (one for each problem).
- Candidate scaling parameters (exponents): Slot `weight` contains a data frame with n combinations of exponents for the external (source) and internal (target) weights

- Stage 2 regressions (after sharing information): Slot `glm.two` contains q lists (one for each problem) of n objects of type `cv.glmnet` (one for each combination of exponents).
- Optimal regularisation parameters: Slot `lambda.min` contains the cross-validated regularisation parameters for the stage 2 regressions.
- Optimal scaling parameters: Slots `weight.ind` and `weight.min` indicate or contain the cross-validated scaling parameters.

References

Armin Rauschenberger, Petr N. Nazarov, and Enrico Glaab (2025). "Estimating sparse regression models in multi-task learning and transfer learning through adaptive penalisation". *Under revision*. https://hdl.handle.net/10993/63425

See Also

Use `coef` to extract coefficients and `predict` to make predictions.

Examples

```
#--- multi-task learning ---
n <- 100
p <- 200
q <- 3

family <- "gaussian"
x <- matrix(data=rnorm(n=n*p),nrow=n,ncol=p)
y <- matrix(data=rnorm(n*q),nrow=n,ncol=q)
object <- sparselink(x=x,y=y,family=family)

#--- transfer learning ---
n <- c(100,50)
p <- 200

x <- lapply(X=n,function(x) matrix(data=stats::rnorm(n*p),nrow=x,ncol=p))
y <- lapply(X=n,function(x) stats::rnorm(x))
family <- "gaussian"
object <- sparselink(x=x,y=y,family=family)
```

Index