Package: smak (via r-universe)

October 24, 2026

Maintainer Steven E. Pav <shabbychef@gmail.com>

Version 0.1.0

Date 2026-10-13

License LGPL-3

Title Scalable Model Averageing Kit

BugReports https://github.com/shabbychef/smak/issues

Description Supports frequentist model averaging on linear regression problems via the scalable method of Zhu et. al. (2022) <doi:10.1080/07350015.2022.2116442>. The scalable model averaging technique projects to singular vectors, performs one linear regression per singular vector, then combines the estimates to optimize either Mallows' criterion, or the jackknife estimate of error.

Depends R (>= 4.0)

Imports generics, methods, quadprog

Suggests broom, testthat, knitr

URL https://github.com/shabbychef/smak

Encoding UTF-8

Collate 'broom.r' 'sirs.r' 'smak.r' 'smamfit.r'

RoxygenNote 7.3.3

NeedsCompilation no

Author Steven E. Pav [aut, cre] (ORCID: <https://orcid.org/0000-0002-4197-6195>)

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

Date/Publication 2026-10-24 11:18:00 UTC

RemoteUrl https://github.com/cran/smak

RemoteRef HEAD

RemoteSha e24929b356d35e7787c6edf962f2eedf46b2a017

Contents

augment.smam 2
coef.smam 3
sirs 4
smak 5
smak-NEWS 7
smam 7
smam.control 9
smamfit 10
tidy.smam 12

Index **14**

`augment.smam` *Augment*

Description

Augment a smam model.

Usage

```
## S3 method for class 'smam'
augment(x, data = NULL, newdata = NULL, se_fit = FALSE, ...)
```

Arguments

`x`	an object of type `smam`
`data`	A base::data.frame or `tibble::tibble()` containing the original data that was used to produce the object `x`. Defaults to `stats::model.frame(x)` so that `augment(my_fit)` returns the augmented original data. **Do not** pass new data to the `data` argument. Augment will report information such as influence and cooks distance for data passed to the `data` argument. These measures are only defined for the original training data.
`newdata`	A `base::data.frame()` or `tibble::tibble()` containing all the original predictors used to create `x`. Defaults to `NULL`, indicating that nothing has been passed to `newdata`. If `newdata` is specified, the `data` argument will be ignored.
`se_fit`	Logical indicating whether or not a `.se.fit` column should be added to the augmented output. For some models, this calculation can be somewhat time-consuming. Defaults to `FALSE`.
`...`	arguments for generic consistency.

Details

Returns a table with predicted fits from a SMA model.

Value

An augmented data frame with added fields:

.fitted The fitted values of the regression model.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

See Also

predict.smam.

---

coef.smam *Extract Coefficients from smam*

Description

Extract Coefficients from smam

Usage

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

Arguments

object	an object for which the extraction of model coefficients is meaningful.
...	Additional arguments passed to or from other methods

Value

A named numeric vector of coefficients.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

sirs *Model-Free Feature Screening*

Description

Performs the model-free feature screening procedure of Zhu et al. for regression problems with many features.

Usage

```
sirs(
  y,
  X,
  thresholding = c("hard", "soft"),
  N = NULL,
  d_prop = 1,
  ties_pragma = c("ignore", "half_value")
)
```

Arguments

y	vector of response data, of length n.
X	a $n \times p$ matrix of independent variables.
thresholding	the method for performing thresholding of the $\hat{w}$ variable to select predictors. For soft thresholding we oversample with noise; for hard thresholding we pick the top N values.
N	the number of variables to select for hard thresholding. Defaults to $n/log(n)$.
d_prop	for soft thresholding controls the number of noise variables added to the set. We add $dpropp$ variables.
ties_pragma	the method for dealing with ties in the y values. For 'ignore' we ignore them and implement the classic value of Zhu et. al. For 'half_value' the indicator function returns one half for tied values in the y.

Details

Performs a model-free screening procedure. First each column of X is z-scored to get zero mean unit variance variables. Then the $\hat{w}$ is computed for each of the columns of X against y. The kth element of this vector is computed as

$$\frac{1}{n(n-1)(n-2)} sum_{1 \leq j \leq n} \left(\sum_{1 \leq i \leq n} X_{i,k} I(y_i < y_j) \right)^2 .$$

Then we select indices k based on the values of this $\hat{w}$, based on the thresholding:

hard Select a fixed number of variates, based on the ordering of $\hat{w}$.

soft Variates are selected if they are larger than the maximum of $\hat{w}$ computed on purely noise variates.

Value

A sorted vector of the indices of which columns of `X` would be selected, with values in $1, 2, \ldots, p$.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

References

Zhu, Liping, Li, Lexin, Li, Runze, and Zhu, Lixing. 2011. "Model-Free Feature Screening for Ultrahigh Dimensional Data." J Am Stat Assoc. 106 (496): 1464-1475. doi:10.1198/jasa.2011.tm10563.

See Also

`smamfit`.

Examples

```
set.seed(1234)
X <- matrix(rnorm(200*500),ncol=500)
beta <- 2 * round(rnorm(ncol(X), sd=0.2)) * runif(ncol(X))
sum(abs(beta) > 0)
true_vars <- which(abs(beta) > 0)
y <- rnorm(nrow(X), mean=X %*% beta)
soft_vals <- sirs(y, X, thresholding='soft')
all(soft_vals %in% true_vars)

hard_vals <- sirs(y, X, thresholding='hard')
all(true_vars %in% hard_vals)
```

---

smak — *scalable model averagking kit*

Description

Scalable Model Averaging.

Frequentist Model Averaging

We consider the basic problem of regressing a scalar y against a vector of independent variables $\vec{x}$. There is a long history of practitioners applying some form of *variable selection* to select a subset of the vector $\vec{x}$ to use in the regression modeling, as a way to avoid overfitting and improve generalization to out-of-sample data. Besides the intractability of testing all 2^p subsets of the p variables in $\vec{x}$, variable selection is often a precursor to questionable statistical practice.

As an alternative to variable selection, Hjort and Claeskens introduced Frequentist Model Averaging. The idea behind FMA is to average together the regression coefficients built on different subsets of the $\vec{x}$, with weights determined typically by one of two methods, Mallows model averaging or Jackknife model averaging.

Scalable Model Averaging

While FMA supplies a stronger theoretical basis than variable selection, it does not solve the computational issues. Enter _Scalable Model Averaging_. Under this technique one first computes the singular value decomposition of the matrix of observed independent variables, X. Then one performs p regressions of the y against each of the left singular values of the X, then performs model averaging on only those p regressions. Zhu *et al.* show how to do this, and prove that the performance is equivalent to the computationally intractable version run on all 2^p regressions.

This package implements that process of performing SVD then regression and weight selection. The regressions produce model fits, but not estimates of uncertainty of the coefficients or total model error. For that one would have to rely on, for example, bootstrapping.

Legal Mumbo Jumbo

smak is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

Note

This package is maintained as a hobby.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

Maintainer: Steven E. Pav <shabbychef@gmail.com> (ORCID)

References

Zhu, Rong, Haiying Wang, Xinyu Zhang, and Hua Liang. 2023. "A Scalable Frequentist Model Averaging Method." Journal of Business & Economic Statistics 41 (4): 1228–37. doi:10.1080/07350015.2022.2116442.

See Also

Useful links:

- https://github.com/shabbychef/smak
- Report bugs at https://github.com/shabbychef/smak/issues

smak-NEWS *News for package 'smak':*

Description

News for package 'smak'

smak Initial Version 0.1.0 (2026-10-13)

- first CRAN release.

smam *Friendly interface to scalable model averaging regression.*

Description

A user friendly interface to the scalable frequentist model averaging regression procedure.

Usage

```
smam(
  formula,
  data,
  weights = NULL,
  na.action = na.omit,
  method = c("jackknife", "mallows"),
  ...
)

## S3 method for class 'smam'
predict(object, newdata, na.action = na.pass, ...)

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

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.

weights	an optional vector of weights, or the string or bare name of the weights in the data for use in the fitting process. Set to NULL for none.
na.action	How to deal with missing values in y, X, wt.
method	the method to be used; for fitting, currently only method = "qr" is supported; method = "model.frame" returns the model frame (the same as with model = TRUE, see below).
...	other arguments.
object	Object of class inheriting from smam.
newdata	a data.frame from which we can extract a model frame via the formula of the object.
x	an object used to select a method.

Details

Performs the scalable model averaging procedure to estimate linear regression of y against X. We find vector of coefficients β such that $X\beta$ is approximately equal to y.

Value

An object of class smam.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

See Also

smamfit.

smam, smamfit.

Examples

```
nobs <- 100
nfeat <- 5
set.seed(1234)
X <- matrix(rnorm(nobs * nfeat),ncol=nfeat)
beta <- rnorm(nfeat)
y <- rnorm(nrow(X), mean=X %*% beta, sd=1)
# now the pretty frontend
data <- cbind(data.frame(outcome=y),as.data.frame(X))

fmla <- outcome ~ V1 + V2 + V3 + V4 + V5
fitm <- smam(fmla,data)

# with weights
data$wts <- runif(nrow(data),min=1,max=2)
fitw <- smam(fmla,data,weights=wts)
```

smam.control *Auxiliary for Controlling SMA Fitting*

Description

Auxiliary function for smam fitting.

Usage

```
smam.control(
  wide_pragma = c("SIRS", "SIRSu", "alpha", "fixed_k", "error"),
  tall_pragma = c("tolerance", "none"),
  alpha = 0.99,
  k = 100
)
```

Arguments

wide_pragma	what to do if X has more columns than rows, see Details.
tall_pragma	what to do if X has as many rows as columns or more, see Details.
alpha	the $alpha$ for the "alpha" wide pragma.
k	the number of singular values kept for the "fixed_k" wide pragma.

Details

The control argument of smam is by default passed to the control argument of smamfit, which uses its selements as arguments to smam.control: the latter provides defaults and sanity checking.

The wide pragma controls how fitting deals with wide matrices X.

SIRS Implements the SIRS procedure of Zhu et al. on X.

SIRSu Implements the SIRS procedure on the left singular values U.

alpha Performs SVD, then keeps the singular values that account for at least α of the total sum of singular values. Always keeps at least one column, however.

fixed_k Keep a fixed number of singular values.

error Throws an error.

The tall pragma controls how fitting deals with tall matrices X.

none Takes all singular values.

tolerance Keeps singular values up to machine epsilon ratio to the largest singular value.

Value

A list.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

References

Zhu, Rong, Haiying Wang, Xinyu Zhang, and Hua Liang. 2023. "A Scalable Frequentist Model Averaging Method." Journal of Business & Economic Statistics 41 (4): 1228–37. doi:10.1080/07350015.2022.2116442.

Zhu, Liping, Li, Lexin, Li, Runze, and Zhu, Lixing. 2011. "Model-Free Feature Screening for Ultrahigh Dimensional Data." J Am Stat Assoc. 106 (496): 1464-1475. doi:10.1198/jasa.2011.tm10563.

See Also

smamfit

---

smamfit — *scalable frequentist model averaging regression fit.*

Description

Fits the scalable model averaging model to a regression problem.

Usage

```
smamfit(
  y,
  X,
  method = c("jackknife", "mallows"),
  wt = NULL,
  sigma2 = NULL,
  Xnames = NULL,
  return.U = FALSE,
  return.X = TRUE,
  return.fitted = TRUE,
  control = list(...),
  formula = NULL,
  ...
)
```

Arguments

y	vector of response data, of length n.
X	a $n \times p$ matrix of independent variables.
method	the selection method for the weights, either Jackknife model averaging or Mallows' model averaging.

`wt`	an optional vector of length n giving the weight of each observation. If given must be non-negative. Weights are replication weights.
`sigma2`	an optional squared variance to be used in Mallows' model averaging. If not given, estimated from the data.
`Xnames`	an optional vector of the names of the columns of X. If null or not given, defaults to `colnames(X)`. Used only for naming the output.
`return.U`	a Boolean for whether the matrix U should be returned in the output.
`return.X`	a Boolean for whether the matrix X should be returned in the output.
`return.fitted`	a Boolean for whether the yhat fitted values should be returned in the output.
`control`	a list of parameters for controlling the the fitting process. This is passed to `smam.control`.
`formula`	an optional formula which will be saved in the object for prediction step later.
`...`	other arguments, which are essentially passed on to `smam.control`, if `control` is not given.

Details

Finds a vector of coefficients β such that $X\beta$ is approximately equal to y. Does this by computing a singular value decomposition of X, then performs p linear regressions of the column of the left singular vectors against y, then takes a weighted combination of those.

Supports both the Mallows Model Averaging and Jackknife Model Averaging approaches.

Value

An object of class `smam`.

Note

smak is free software.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

References

Zhu, Rong, Haiying Wang, Xinyu Zhang, and Hua Liang. 2023. "A Scalable Frequentist Model Averaging Method." Journal of Business & Economic Statistics 41 (4): 1228–37. doi:10.1080/07350015.2022.2116442.

See Also

the friendly interface `smam`.

Examples

```
nobs <- 100
nfeat <- 5
set.seed(1234)
X <- matrix(rnorm(nobs * nfeat),ncol=nfeat)
beta <- rnorm(nfeat)
eta <- X %*% beta
y <- rnorm(nrow(X), mean=X %*% beta, sd=1)

mod0 <- smamfit(y=y,X=X)
summary(mod0)
```

---

tidy.smam *Tidy*

Description

Tidy a smam model.

Usage

```
## S3 method for class 'smam'
tidy(x, ...)
```

Arguments

x	an object of type smam.
...	arguments for generic consistency.

Details

Returns a table with information on the fit coefficients from the scalable model averaging model. At the moment, no standard errors are computed.

Value

A tidy data frame with fields

term The name of the estimated parameter. Betas typically come before gammas.

estimate The estimated parameter.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

Examples

```
nobs <- 100
nfeat <- 5
set.seed(1234)
X <- matrix(rnorm(nobs * nfeat),ncol=nfeat)
beta <- rnorm(nfeat)
eta <- X %*% beta
y <- rnorm(nrow(X), mean=X %*% beta, sd=1)

mod0 <- smamfit(y=y,X=X)
print(tidy(mod0))
```

Index

∗ **fitting**
    smam, 7
    smamfit, 10
∗ **package**
    smak, 5

as.data.frame, *7*
augment.smam, 2

base::data.frame, *2*
base::data.frame(), *2*

coef.smam, 3

formula, *7*

predict.smam, *3*
predict.smam *(smam)*, 7
print.smam *(smam)*, 7

sirs, 4
smak, 5
smak-NEWS, 7
smak-package *(smak)*, 5
smam, 7, *8*, *11*
smam.control, 9, *11*
smamfit, *5*, *8*, 10, *10*

tibble::tibble(), *2*
tidy.smam, 12