Package: wflsa (via r-universe)

November 21, 2024

Title Weighted Fused LASSO Signal Approximator ('wFLSA')

Version 1.2.1

Author Louis Dijkstra [aut, cre]

Maintainer Louis Dijkstra <dijkstra@leibniz-bips.de>

Description A package for computing the Weighted Fused LASSO Signal Approximator (wFLSA).

License GPL (>= 3)

URL https://github.com/bips-hb/wflsa, https://bips-hb.github.io/wflsa/

BugReports https://github.com/bips-hb/wflsa/issues

Depends R (>= 4.0.0)

Imports Rcpp

LinkingTo Rcpp

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Roxygen list(markdown = TRUE)

Repository https://bips-hb.r-universe.dev

RemoteUrl https://github.com/bips-hb/wflsa

RemoteRef HEAD

RemoteSha bb729f8d0ca65283a500c6beedfdb717f135331c

Contents

`calculate_diagonal_matrix_A` *Determine the Diagonal of Matrix A*

Description

The wFLSA algorithm requires the choice of a matrix A such that $A - D'D$ is positive semidefinite. We choose the matrix A to be diagonal with a fixed value a. This function determines the smallest value of a such that $A - D'D$ is indeed positive semidefinite. We do this be determining the largest eigenvalue

Usage

```
calculate_diagonal_matrix_A(W, eta1, eta2)
```

Arguments

`W`	Weight matrix W
`eta1, eta2`	The values $\eta_1 = \lambda_1/\rho$ and $\eta_2 = \lambda_2/\rho$

Value

Value of a

References

Zhu, Y. (2017). An Augmented ADMM Algorithm With Application to the Generalized Lasso Problem. Journal of Computational and Graphical Statistics, 26(1), 195–204. https://doi.org/10.1080/10618600.2015.111449

`genlassoRcpp` *Solving Generalized LASSO with fixed $\lambda = 1$*

Description

Solves efficiently the generalized LASSO problem of the form

$$\hat{\beta} = \text{argmin} \, \frac{1}{2}||y - \beta||_2^2 + ||D\beta||_1$$

where β and y are m-dimensional vectors and D is a $(c \times m)$-matrix where $c \geq m$. We solve this optimization problem using an adaption of the ADMM algorithm presented in Zhu (2017).

Usage

```
genlassoRcpp(Y, W, m, eta1, eta2, a, rho, max_iter, eps, truncate)
```

Arguments

Y	The y vector of length m
W	The weight matrix W of dimensions $m \times m$
m	The number of graphs
eta1	Equals λ_1/rho
eta2	Equals λ_2/rho
a	Value added to the diagonal of $-D'D$ so that the matrix is positive definite, see `matrix_A_inner_ADMM` in package `CVN`
rho	The ADMM's parameter
max_iter	Maximum number of iterations
eps	Stopping criterion. If differences are smaller than ϵ, algorithm is halted
truncate	Values below `truncate` are set to `0`

Value

The estimated vector $\hat{\beta}$

References

Zhu, Y. (2017). An Augmented ADMM Algorithm With Application to the Generalized Lasso Problem. Journal of Computational and Graphical Statistics, 26(1), 195–204. https://doi.org/10.1080/10618600.2015.111449

---

print.wflsa.fit — *Print Function for the Weighted Fused LASSO Signal Approximator (wFLSA) fit object*

Description

This function prints information about the fitted wFLSA object, including the number of variables, the number of lambda pairs, and the estimated beta coefficients.

Usage

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

Arguments

x	An object of class wflsa.fit.
...	Additional arguments to be passed to the print function.

`wflsa` *The Weighted Fused Lasso Signal Approximator (wFLSA) Algorithm*

Description

Solves the weighted Fused LASSO Signal Approximator optimization problem using an ADMM-based approach. The problem is formulated as follows:

$$\hat{\beta} = \text{argmin} \, \frac{1}{2}||y - \beta||_2^2 + \lambda_1||\beta||_1 + \lambda_2 \sum_{i<j} w_{ij}|\beta_i - \beta_j|$$

where:

- y is the response with mean 0.
- β is the vector of coefficients to be estimated.
- $|| \cdot ||_1$ and $|| \cdot ||_2$ are the L_1- and L_2-norms, respectively.
- $\lambda_1 > 0$ is the regularization parameter controlling the strength of the sparsity penalty.
- $\lambda_2 > 0$ is the regularization parameter controlling the smoothness.
- $w_{ij} \in [0, 1]$ is the weight between the i-th and j-th coefficient.

Usage

```
wflsa(
  y,
  W,
  lambda1 = c(0.1),
  lambda2 = c(0.1),
  rho = 1,
  max_iter = 1e+05,
  eps = 1e-10,
  truncate = 1e-04,
  offset = TRUE
)
```

Arguments

`y`	Vector of length p representing the response variable (assumed to be centered).
`W`	Weight matrix of dimensions $p \times p$.
`lambda1`	Vector of positive regularization parameters for L_1 penalty.
`lambda2`	Vector of positive regularization parameters for smoothness penalty.
`rho`	ADMM's parameter (Default: `1`).
`max_iter`	Maximum number of iterations (Default: `1e5`).
`eps`	Stopping criterion. If differences are smaller than `eps`, the algorithm halts (Default: `1e-10`).
`truncate`	Values below `truncate` are set to `0` (Default: `1e-4`).
`offset`	Logical indicating whether to include an intercept term (Default: `TRUE`).

Value

A list containing:

- `betas`: Estimated vector $\hat{\beta}$ from the Weighted Fused LASSO.
- `tuning_parameters`: Data frame with tuning parameters. The column `df` contains the number of non-zero coefficients for the different lambda-values
- all input variables.

Note

Important Note: The algorithm assumes y to be centered, i.e., its mean is 0.

See Also

`genlassoRcpp()`

Examples

```
# Example usage of the wflsa function
y <- c(1, 2, 3)
W <- matrix(c(1, 0, 0, 0, 1, 0, 0, 0, 1), ncol = 3)
lambda1 <- c(0.1, 0.2)
lambda2 <- c(0.1, 0.2)
result <- wflsa(y, W, lambda1, lambda2)
```

Index