Package ‘RKHSMetaMod’

July 7, 2019

Type Package
Title Ridge Group Sparse Optimization Problem for Estimation of a Meta Model Based on Reproducing Kernel Hilbert Spaces
Version 1.1
Date 2019-07-06
Author Halaleh Kamari
Maintainer Halaleh Kamari <halaleh.kamari@univ-evry.fr>
License GPL (>= 2)
Imports Rcpp (>= 1.0.0)
Suggests lhs
LinkingTo Rcpp, RcppEigen, RcppGSL
NeedsCompilation yes
Repository CRAN
Date/Publication 2019-07-06 22:10:04 UTC

R topics documented:

RKHSMetaMod-package .................................................. 2
calc_Kv ............................................................... 3
grplasso_q .......................................................... 5
mu_max ............................................................... 7
pen_MetMod ......................................................... 8
PredErr ............................................................... 11
RKHSgrplasso ......................................................... 13
RKHSMetMod ......................................................... 15
RKHSMetMod_qmax .................................................... 17
SI_emp ............................................................... 20

Index 22
Set of Rcpp and R functions to produce a sequence of meta models that are the solutions of the RKHS Ridge Group Sparse or RKHS Group Lasso optimization problems, calculate their associated prediction errors as well as their empirical sensitivity indices.

Description

Fits a meta model to an unknown model \( m \) by solving the ridge group sparse (or group lasso) optimization problem based on the reproducing kernel Hilbert spaces (RKHS), for the Gaussian regression model:

\[
Y = m(X) + \sigma \varepsilon,
\]

where variables \( X_1, \ldots, X_d \) are independent and uniformly distributed on \([0, 1]\) and are independent of \( \varepsilon \)'s.

We define the ridge group sparse criteria by:

\[
C(f_0, \theta) = \|Y - f_0 I_n - \sum_{v \in P} K_v \theta_v \|^2 + \sqrt{n} \gamma \sum_{v \in P} \|K_v \theta_v \| + n \mu \sum_{v \in P} \|K_v^{1/2} \theta_v \|,
\]

and the group lasso criteria is obtained by setting \( \gamma = 0 \) in the criteria above. We set \( \mu_g = \sqrt{n} \mu \) to be the group lasso penalty parameter.

For each pair of the penalty parameters \((\mu, \gamma)\) in the ridge group sparse criteria, one meta model, called RKHS meta model, is calculated. The RKHS meta model is an additive model with at most \( v_{\text{Max}} \) groups. It satisfies the properties of the Hoeffding decomposition, and its terms estimate the terms in the Hoeffding decomposition of the function \( m \).

These estimators are evaluated using a testing dataset. That is, the prediction error is calculated for each RKHS meta model and the one with the minimum prediction error is the "best" estimator for the true model \( m \). It provides a function that estimates the empirical sensitivity indices of the "best" RKHS meta model as an approximation of the true sensitivity indices.

Details

Details.

Author(s)

Halaleh Kamari.

Maintainer: <halaleh.kamari@univ-evry.fr>

References

Function to calculate the Gram matrices and their eigenvalues and
eigenvectors for a chosen reproducing kernel.

Description

Calculates the Gram matrices $K_v$ for $v = 1, \ldots, \text{vMax}$, and returns their associated eigenvalues and
eigenvectors. The calculated Gram matrices may be not positive definite. The option "correction" of this function allows to replace the matrices $K_v$ that are not positive definite by their "nearest positive definite" matrices.

Usage

calc_Kv(X, kernel, Dmax, correction, verbose, tol)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Matrix of observations with $n$ rows and $d$ columns.</td>
</tr>
<tr>
<td>kernel</td>
<td>Character, the type of the reproducing kernel: matern (matern kernel), brownian (brownian kernel), gaussian (gaussian kernel), linear (linear kernel), quad (quadratic kernel).</td>
</tr>
<tr>
<td>Dmax</td>
<td>Integer, between 1 and $d$, indicates the order of interactions considered in the meta model: Dmax= 1 is used to consider only the main effects, Dmax= 2 to include the main effects and the interactions of order 2, ...</td>
</tr>
<tr>
<td>correction</td>
<td>Logical, if TRUE, the program makes the correction to the matrices $K_v$ that are not positive definite (see details). Set as TRUE by default.</td>
</tr>
<tr>
<td>verbose</td>
<td>Logical, if TRUE, the group $v$ for which the correction is done is printed. Set as TRUE by default.</td>
</tr>
</tbody>
</table>
Scalar, used if correction is TRUE. For each matrix $K_v$ if $\lambda_{min} < \lambda_{max} \times \text{tol}$, then the correction to $K_v$ is done (see details). Set as $1e^{-8}$ by default.

Details

Let $\lambda_{v,i}, i = 1, ..., n$ be the eigenvalues associated with matrix $K_v$. Set $\lambda_{max} = \max_i \lambda_{v,i}$ and $\lambda_{min} = \min_i \lambda_{v,i}$. The eigenvalues of $K_v$ that is not positive definite are replaced by $\lambda_{v,i} + \text{epsilon}$, with $\text{epsilon} = \lambda_{max} \times \text{tol}$. The value of $\text{tol}$ depends on the type of the kernel and it is chosen small.

Value

List of two components "names.Grp" and "kv":

- names.Grp: Vector of size vMax, indicates the name of groups included in the meta model.
- kv: List of vMax components with the same names as the vector names.Grp. Each element of the list is a list of two components "Evalues" and "Q":
  - Evalues: Vector of size $n$, eigenvalues of each Gram matrix $K_v$.
  - Q: Matrix with $n$ rows and $n$ columns, eigenvectors of each Gram matrix $K_v$.

Note

Note.

Author(s)

Halaleh Kamari

References


See Also

RKHSMetaMod

Examples

d <- 3
n <- 50
library(lhs)
X <- maximinLHS(n, d)
c <- c(0.2, 0.6, 0.8)
F <- 1; for (a in 1:d) F <- F*(abs(4*X[,a]-2)+c[a])/(1+c[a])
epsilon <- rnorm(n,0,1); sigma <- 0.2
Y <- F + sigma*epsilon
Dmax <- 3
kernel <- "matern"
Kv <- calc_Kv(X, kernel, Dmax)
Function to fit a solution with q active groups of an RKHS Group Lasso problem.

Description

Fits a solution of the group lasso problem based on RKHS, with q active groups in the obtained solution for the Gaussian regression model. It determines $\mu_g(q)$, for which the number of active groups in the solution of the RKHS group lasso problem is equal to q, and returns the RKHS meta model associated with $\mu_g(q)$.

Usage

`grplasso_q(Y, Kv, q, rat, Num)`

Arguments

- **Y**: Vector of response observations of size n.
- **Kv**: List of eigenvalues and eigenvectors of positive definite Gram matrices $K_v$ and their associated group names. It should have the same format as the output of the function `calc_Kv` (see details).
- **q**: Integer, the number of active groups in the obtained solution.
- **rat**: Positive scalar, used to restrict the minimum value of $\mu_g$, to be evaluated in the RKHS Group Lasso algorithm, $\mu_{\text{min}} = \mu_{\text{max}}/\text{rat}$. The value $\mu_{\text{max}}$ is calculated inside the program, see function `mu_max`.
- **Num**: Integer, used to restrict the number of different values of the penalty parameter $\mu_g$ to be evaluated in the RKHS Group Lasso algorithm, until it achieves $\mu_g(q)$: for Num = 1 the program is done for 3 values of $\mu_g$, $\mu_1 = (\mu_{\text{min}} + \mu_{\text{max}})/2$, $\mu_2 = (\mu_{\text{min}} + \mu_1)/2$ or $\mu_2 = (\mu_1 + \mu_{\text{max}})/2$ depending on the value of q associated with $\mu_1$, $\mu_3 = \mu_{\text{min}}$.

Details

Input Kv should contain the eigenvalues and eigenvectors of positive definite Gram matrices $K_v$. It is necessary to set input "correction" in the function `calc_Kv` equal to "TRUE".

Value

List of 4 components: "mus", "qs", "mu", "res":

- **mus**: Vector, values of the evaluated penalty parameters $\mu_g$ in the RKHS group lasso algorithm until it achieves $\mu_g(q)$.
- **qs**: Vector, number of active groups associated with each value of $\mu_g$ in mus.
Scalar, value of $\mu_g(q)$.

Scalar, estimated value of intercept.

Matrix with vMax rows and $n$ columns. Each row of the matrix is the estimated vector $\theta_v$ for $v = 1, ..., vMax$.

Matrix with $n$ rows and vMax columns. Each row of the matrix is the estimated value of $f_v = K_v\theta_v$.

Vector of size $n$, indicates the estimator of $m$.

Vector of size vMax, estimated values of the penalty norm.

Vector of active groups.

Vector of the names of the active groups.

Scalar, equals to $\|Y - f_0 - \sum_v K_v\theta_v\|^2$.

Scalar, indicates the value of the penalized criteria.

Integer, number of iterations until convergence is reached.

TRUE or FALSE. Indicates whether the algorithm has converged or not.

Scalar, value of the first convergence criteria at the last iteration, $\frac{\text{crit}_{\text{lastIter}} - \text{crit}_{\text{lastIter}-1}}{\text{crit}_{\text{lastIter}-1}}$.

Scalar, value of the second convergence criteria at the last iteration, $\frac{\|\theta_{\text{lastIter}} - \theta_{\text{lastIter}-1}\|^2}{\theta_{\text{lastIter}-1}}$.

Note

Note.

Author(s)

Halaleh Kamari

References


See Also

calc_Kv, mu_max

Examples

d <- 3
n <- 50
library(lhs)
X <- maximinLHS(n, d)
c <- c(0.2, 0.6, 0.8)
F <- 1; for (a in 1:d) F <- F*(abs(4*X[,a]-2)*c[a])/(1+c[a])
epsilon <- rnorm(n, 0, 1); sigma <- 0.2
Function to find the maximal value of the penalty parameter in the RKHS Group Lasso problem.

Description
Calculates the value of the penalty parameter in the RKHS group lasso problem when the first penalized parameter group enters the model.

Usage
mu_max(Y, matZ)

Arguments
Y Vector of response observations of size n.
matZ List of vMax components. Each component includes the eigenvalues and eigenvectors of the positive definite Gram matrices $K_v, v = 1,...,vMax$. It should have the same format as the output "kv" of the function calc_Kv.

Details
Details.

Value
An object of type numeric is returned.

Note
Note.

Author(s)
Halaleh Kamari
References


See Also

calc_Kv

Examples

d <- 3
n <- 50
library(lhs)
X <- maximinLHS(n, d)
c <- c(0.2, 0.6, 0.8)
F <- 1; for (a in 1:d) F <- F*(abs(4*X[,a]-2)+c[a])/(1+c[a])
epsilon <- rnorm(n, 0, 1); sigma <- 0.2
Y <- F + sigma*epsilon
Dmax <- 3
kernel <- "matern"
Kv <- calc_Kv(X, kernel, Dmax, TRUE, TRUE)
matZ <- Kv$K
mumax <- mu_max(Y, matZ)
mumax

pen_MetMod

Function to fit a solution of the RKHS Ridge Group Sparse problem.

Description

Fits the solution of the RKHS ridge group sparse optimization problem for the Gaussian regression model.

Usage

pen_MetMod(Y, Kv, gamma, mu, resg, gama_v, mu_v, maxIter, verbose, calcStwo)

Arguments

Y Vector of response observations of size n.
Kv List, includes the eigenvalues and eigenvectors of the positive definite Gram matrices $K_v, v = 1, ..., VMax$ and their associated group names. It should have the same format as the output of the function calc_Kv (see details).
gamma Vector of positive scalars. Values of the penalty parameter $\gamma$ in decreasing order.
**pen_MetMod**

- **mu**
  - Vector of positive scalars. Values of the penalty parameter $\mu$ in decreasing order.

- **resg**
  - List of initial parameters, includes the `RHSSgrplasso` objects for each value of the penalty parameter $\mu$.

- **gama_v**
  - Scalar zero or vector of vMax positive scalars, considered as weights for the Ridge penalty. Set to zero, to consider no weights, i.e. all weights equal to 1.

- **mu_v**
  - Scalar zero or a vector with vMax scalars, considered as weights of Sparse Group penalty. Set to zero, to consider no weights, i.e. all weights equal to 1.

- **maxIter**
  - Integer, shows the maximum number of loops through initial active groups at the first step and maximum number of loops through all groups at the second step. Set as 1000 by default.

- **verbose**
  - Logical, if TRUE, for each pair of penalty parameters $(\mu, \gamma)$ it prints: the number of current iteration, active groups and convergence criterias. Set as FALSE by default.

- **calcStwo**
  - Logical, if TRUE, the program does a second step after convergence: the algorithm is done over all groups by taking the estimated parameters at the first step as initial values. Set as FALSE by default.

**Details**

Input $K_v$ should contain the eigenvalues and eigenvectors of positive definite Gram matrices $K_v$. It is necessary to set input "correction" in the function `calc_Kv` equal to "TRUE".

**Value**

List of l components, with l equals to the number of pairs of the penalty parameters $(\mu, \gamma)$. Each component of the list is a list of 3 components "mu", "gamma" and "Meta-Model":

- **mu**
  - Positive scalar, an element of the input vector mu associated with the estimated Meta-Model.

- **gamma**
  - Positive scalar, an element of the input vector gamma associated with the estimated Meta-Model.

- **Meta-Model**
  - Estimated meta model associated with penalty parameters mu and gamma. List of 16 components:
    - **intercept**
      - Scalar, estimated value of intercept.
    - **teta**
      - Matrix with vMax rows and $n$ columns. Each row of the matrix is the estimated vector $\theta_v$ for $v = 1, ..., vMax$.
    - **fit.v**
      - Matrix with $n$ rows and vMax columns. Each row of the matrix is the estimated value of $f_v = K_v \theta_v$.
    - **fitted**
      - Vector of size $n$, indicates the estimator of $m$.
    - **Norm.n**
      - Vector of size vMax, estimated values for the Ridge penalty norm.
    - **Norm.H**
      - Vector of size vMax, estimated values for the Group Sparse penalty norm.
    - **supp**
      - Vector of active groups.
    - **Nsupp**
      - Vector of the names of the active groups.
Scalar equals to \[ \| Y - f_0 - \sum_v K_v \theta_v \|^2. \]

Scalar indicates the value of the penalized criteria.

Vector of size vMax, coefficients of the Ridge penalty norm, \( \sqrt{n} \gamma \times \text{gamma}_v \).

Vector of size vMax, coefficients of the Group Sparse penalty norm, \( n \mu \times \text{mu}_v \).

List of three components if calcStwo=TRUE (two components if calcStwo=FALSE): maxIter, number of iterations until convergence is reached at first step and the number of iterations until convergence is reached at second step (maxIter, and the number of iterations until convergence is reached at first step).

TRUE or FALSE. Indicates whether the algorithm has converged or not.

List of two components if calcStwo=TRUE (one component if calcStwo=FALSE): value of convergence criteria at the last iteration of each step, \[ \frac{\| \theta_{\text{lastIter}} - \theta_{\text{lastIter}-1} \|}{\| \theta_{\text{lastIter}-1} \|}. \]

List of two components if calcStwo=TRUE (one component if calcStwo=FALSE): value of convergence criteria at the last iteration, \[ \frac{\text{crit}_{\text{lastIter}} - \text{crit}_{\text{lastIter}-1}}{\text{crit}_{\text{lastIter}-1}} \] of each step.

Note

Note.

Halaleh Kamari

References


See Also

calc_Kv, RKHSGrpllasso

Examples

d <- 3
n <- 50
library(lhs)
X <- maximinLHS(n, d)
c <- c(0.2, 0.6, 0.8)
F <- !for (a in 1:d) F <- F*(abs(4*X[,a]-2)+c[a])/(1+c[a])
epsilon <- rnorm(n,0,1); sigma <- 0.2
Y <- F + sigma*epsilon
Dmax <- 3
kernel <- "matern"
Kv <- calc_Kv(X, kernel, Dmax, TRUE, TRUE, tol = 1e-08)
**Description**

Computes the prediction error by considering a testing dataset.

**Usage**

```r
PredErr(X, XT, YT, mu, gamma, res, kernel, Dmax)
```

**Arguments**

- **X** Matrix of observations with \( n \) rows and \( d \) columns.
- **XT** Matrix of observations of the testing dataset with \( n_{test} \) rows and \( d \) columns.
- **YT** Vector of response observations of testing dataset of size \( n_{test} \).
- **mu** Vector of positive scalars. Values of the Group Sparse penalty parameter in decreasing order. See function `RKHSmetMod`.
- **gamma** Vector of positive scalars. Values of the Ridge penalty parameter in decreasing order. See function `RKHSmetMod`.
- **res** List, includes a sequence of estimated meta models for the learning dataset, using RKHS Ridge Group Sparse or RKHS Group Lasso algorithm, associated with the penalty parameters `mu` and `gamma`. It should have the same format as the output of one of the functions: `pen_MetMod`, `RKHSmetMod` or `RKHSmetMod_qmax`.
- **kernel** Character, shows the type of the reproducing kernel: matern, brownian, gaussian, linear, quad. The same kernel should be chosen as the one used for the learning dataset. See function `calc_Kv`.
**Dmax**

Integer between 1 and \(d\). The same Dmax should be chosen as the one used for learning dataset. See function `calc_Kv`.

**Details**

Details.

**Value**

Matrix of the prediction errors is returned. Each element of the matrix is the obtained prediction error associated with one RKHS meta model in "res".

**Note**

Note.

**Author(s)**

Halaleh Kamari

**References**


**See Also**

calc_Kv, pen_MetMod, RKHSMetMod, RKHSMetMod_qmax

**Examples**

```r
Dmax <- 3
n <- 50
nT <- 50
library(lhs)
x <- maximinLHS(n, d)
xT <- maximinLHS(nT, d)
c <- c(0.2, 0.6, 0.8)
F <- 1; for (a in 1:d) F <- F*(abs(4*X[,a]-2)+c[a])/(1+c[a])
FT <- 1; for (a in 1:d) FT <- FT*(abs(4*xT[,a]-2)+c[a])/(1+c[a])
sigma <- 0.2
epsilon <- rnorm(n, 0, 1); 
y <- F + sigma*epsilon
epsilonT <- rnorm(nT, 0, 1); YT <- FT + sigma*epsilonT
Dmax <- 3
kernel <- "matern"
frc <- c(10, 100)
gamma <- c(0.5, 0.01, 0.001)
res <- RKHSMetMod(Y, X, kernel, Dmax, gamma, frc, FALSE)
u <- vector()
1 <- length(gamma)
for(i in 1:length(frc)){u[i]=res[[i-1]*l+1]$
```
error <- PredErr(X, XT, YT, mu, gamma, res, kernel, Dmax)
error

---

Function to fit a solution of an RKHS Group Lasso problem.

Description

Fits the solution of an RKHS group lasso problem for the Gaussian regression model.

Usage

RKHSgrplasso(Y, Kv, mu, maxIter, verbose)

Arguments

Y  Vector of response observations of size \( n \).
Kv List, includes the eigenvalues and eigenvectors of the positive definite Gram matrices \( K_v \), \( v = 1, ..., \text{vMax} \) and their associated group names. It should have the same format as the output of the function calc_Kv (see details).
mu Positive scalar, value of the penalty parameter \( \mu_g \) in the RKHS Group Lasso problem.
maxIter Integer, shows the maximum number of loops through all groups. Set as 1000 by default.
verbose Logical, if TRUE, prints: the number of current iteration, active groups and convergence criterias. Set as FALSE by default.

Details

Input Kv should contain the eigenvalues and eigenvectors of positive definite Gram matrices \( K_v \). It is necessary to set input correction in the function calc_Kv equal to "TRUE".

Value

Estimated RKHS meta model, list with 13 components:

- intercept Scalar, estimated value of intercept.
- teta Matrix with vMax rows and \( n \) columns. Each row of the matrix is the estimated vector \( \theta_v \) for \( v = 1, ..., \text{vMax} \).
- fit.v Matrix with \( n \) rows and vMax columns. Each row of the matrix is the estimated value of \( f_v = K_v \theta_v \).
- fitted Vector of size \( n \), indicates the estimator of \( m \).
- Norm.H Vector of size vMax, estimated values of the penalty norm.
- supp Vector of active groups.
- Nsupp Vector of the names of the active groups.
SCR Scalar, equals to \( \| Y - f_0 - \sum_v K_v \theta_v \|^2 \).

crit Scalar, indicates the value of penalized criteria.

MaxIter Integer, number of iterations until convergence is reached.

critScalar, indicates the value of penalized criteria.

convergence TRUE or FALSE. Indicates whether the algorithm has converged or not.

RelDiffCrit Scalar, value of the first convergence criteria at the last iteration, \( \frac{\text{crit}_{\text{lastIter}} - \text{crit}_{\text{lastIter-1}}}{\text{crit}_{\text{lastIter-1}}} \).

RelDiffPar Scalar, value of the second convergence criteria at the last iteration, \( \| \frac{\theta_{\text{lastIter}} - \theta_{\text{lastIter-1}}}{\theta_{\text{lastIter-1}}} \|^2 \).

Note

Note.

Author(s)

Halaleh Kamari

References


See Also

calc_Kv

Examples

d <- 3
n <- 50
library(lhs)
X <- maximinLHS(n, d)
c <- c(0.2, 0.6, 0.8)
F <- 1; for (a in 1:d) F <- F*(abs(4*X[,a]-2)+c[a])/(1+c[a])
epsilon <- rnorm(n, 0, 1); sigma <- 0.2
Y <- F + sigma*epsilon
Dmax <- 3
kernel <- "matern"
Kv <- calc_Kv(X, kernel, Dmax, TRUE, TRUE)
matZ <- Kv$Kv
mumax <- mu_maxHyL matZI
mug <- mumax/QP
gr <- RKHSgrplasso(Y, Kv, mug, 1000, FALSE)
gr$Nsupp
**RKHSMetMod**

*Function to produce a sequence of meta models that are the solutions of the RKHS Ridge Group Sparse or RKHS Group Lasso optimization problems.*

**Description**

Calculates the Gram matrices $K_v$ for a chosen reproducing kernel and fits the solution of an RKHS ridge group sparse or an RKHS group lasso problem for each pair of penalty parameters $(\mu, \gamma)$, for the Gaussian regression model.

**Usage**

```r
RKHSMetMod(Y, X, kernel, Dmax, gamma, frc, verbose)
```

**Arguments**

- `Y`  Vector of response observations of size $n$.
- `X`  Matrix of observations with $n$ rows and $d$ columns.
- `kernel`  Character, indicates the type of the reproducing kernel: matern (matern kernel), brownian (brownian kernel), gaussian (gaussian kernel), linear (linear kernel), quad (quadratic kernel). See the function `calc Kv`.
- `Dmax`  Integer, between 1 and $d$, indicates the order of interactions considered in the meta model: $D_{max} = 1$ is used to consider only the main effects, $D_{max} = 2$ to include the main effects and the interactions of order 2, ... . See the function `calc Kv`.
- `gamma`  Vector of non negative scalars, values of the penalty parameter $\gamma$ in decreasing order. If $\gamma = 0$ the function solves an RKHS Group Lasso problem and for $\gamma > 0$ it solves an RKHS Ridge Group Sparse problem.
- `frc`  Vector of positive scalars. Each element of the vector sets a value to the penalty parameter $\mu$, $\mu = \mu_{max}/(\sqrt{n} \times frc)$. The value $\mu_{max}$ is calculated by the program. See the function `mu_max`.
- `verbose`  Logical, if TRUE, prints: the group $v$ for which the correction of Gram matrix $K_v$ is done, and for each pair of the penalty parameters $(\mu, \gamma)$: the number of current iteration, active groups and convergence criterias. Set as FALSE by default.

**Details**

Details.
Value
List of 1 components, with 1 equals to the number of pairs of the penalty parameters \((\mu, \gamma)\). Each component of the list is a list of 3 components "mu", "gamma" and "Meta-Model":

- mu: Positive scalar, penalty parameter \(\mu\) associated with the estimated Meta-Model.
- gamma: Positive scalar, an element of the input vector gamma associated with the estimated Meta-Model.
- Meta-Model: An RKHS Ridge Group Sparse or RKHS Group Lasso object associated with the penalty parameters mu and gamma:
  - intercept: Scalar, estimated value of intercept.
  - teta: Matrix with vMax rows and \(n\) columns. Each row of the matrix is the estimated vector \(\theta_v\) for \(v = 1, \ldots, v\text{Max}\).
  - fit.Nv: Matrix with \(n\) rows and vMax columns. Each row of the matrix is the estimated value of \(f_v = K_v \theta_v\).
  - fitted: Vector of size \(n\), indicates the estimator of \(m\).
  - Norm.n: Vector of size vMax, estimated values for the Ridge penalty norm.
  - Norm.H: Vector of size vMax, estimated values of the Sparse Group penalty norm.
  - supp: Vector of active groups.
  - Nsupp: Vector of the names of the active groups.
  - SCR: Scalar, equals to \(\|Y - f_0 - \sum_v K_v \theta_v\|_2^2\).
  - crit: Scalar, indicates the value of the penalized criteria.
  - gamma.v: Vector of size vMax, coefficients of the Ridge penalty norm, \(\sqrt{n} \gamma \times \text{gama}_v\).
  - mu.v: Vector of size vMax, coefficients of the Group Sparse penalty norm, \(n \mu \times \text{mu}_v\).
- iter: List of two components: maxIter, and the number of iterations until the convergence is achieved.
- convergence: TRUE or FALSE. Indicates whether the algorithm has converged or not.
- RelDiffCrit: Scalar, value of the first convergence criteria at the last iteration, \(\frac{\|\theta_{\text{lastIter}} - \theta_{\text{lastIter-1}}\|_2^2}{\theta_{\text{lastIter-1}}}\).
- RelDiffPar: Scalar, value of the second convergence criteria at the last iteration, \(\frac{\text{crit}_{\text{lastIter}} - \text{crit}_{\text{lastIter-1}}}{\text{crit}_{\text{lastIter-1}}}\).

Note
For the case \(\gamma = 0\) the outputs "mu"= \(\mu_g\) and "Meta-Model" is the same as the one returned by the function RKHSgrplasso.

Author(s)
Halaleh Kamari

References
RKHSMetMod_qmax

Function to produce a sequence of meta models, with at most qmax active groups in each meta model. The meta models are the solutions of the RKHS Ridge Group Sparse or RKHS Group Lasso optimization problems.

Description

Calculates the Gram matrices $K_v$ for a chosen kernel, determines $\mu$, note $\mu(qmax)$, for which the number of active groups in the RKHS group lasso solution is equal to qmax, and fits a solution of an RKHS ridge group sparse or an RKHS group lasso problem for each pair of penalty parameters $(\mu(qmax), \gamma)$, in the Gaussian regression model.

Usage

RKHSMetMod_qmax(Y, X, kernel, Dmax, gamma, qmax, rat, Num, verbose)

Arguments

Y Vector of response observations of size n.
X Matrix of observations with n rows and d columns.
kernel Character, indicates the type of the reproducing kernel: matern (matern kernel), brownian (brownian kernel), gaussian (gaussian kernel), linear (linear kernel), quad (quadratic kernel). See the function calc_Kv.
**Dmax**
Integer, between 1 and \(d\), indicates the order of interactions considered in the meta model: \(\text{Dmax} = 1\) is used to consider only the main effects, \(\text{Dmax} = 2\) to include the main effects and the interactions of order 2, \ldots. See the function `calc_kv`.

**gamma**
Vector of non negative scalars, values of the penalty parameter \(\gamma\) in decreasing order. If \(\gamma = 0\) the function solves an RKHS Group Lasso problem and for \(\gamma > 0\) it solves an RKHS Ridge Group Sparse problem.

**qmax**
Integer, shows the maximum number of active groups in the obtained solution.

**rat**
Positive scalar, to restrict the minimum value of \(\mu\) considered in the algorithm, \(\mu_{\text{min}} = \mu_{\text{max}} / (\sqrt{n} \times \text{rat})\). The value \(\mu_{\text{max}}\) is calculated inside the program, see function `mu_max`.

**Num**
Integer, it is used to restrict the number of different values of the penalty parameter \(\mu\) to be evaluated in the RKHS Group Lasso algorithm until it achieves \(\mu(qmax)\): for \(\text{Num} = 1\) the program is done for 3 different values of \(\mu\), \(\mu_1 = (\mu_{\text{min}} + \mu_{\text{max}}) / 2\), \(\mu_2 = (\mu_{\text{min}} + \mu_1) / 2\) or \(\mu_2 = (\mu_1 + \mu_{\text{max}}) / 2\) depending on the number of active groups in the meta model associated with \(\mu_1\), \(\mu_3 = \mu_{\text{min}}\).

**verbose**
Logical, if TRUE, prints: the group \(v\) for which the correction of Gram matrix \(K_v\) is done, and for each pair of \((\mu, \gamma)\): the number of current iteration, active groups and convergence criterias. Set as FALSE by default.

**Details**
Details.

**Value**
List of three components "mus", "qs", and "MetaModel":

<table>
<thead>
<tr>
<th>mus</th>
<th>Vector, values of the evaluated penalty parameters (\mu) in the RKHS Group Lasso algorithm until it achieves (\mu(qmax)).</th>
</tr>
</thead>
<tbody>
<tr>
<td>qs</td>
<td>Vector, number of active groups associated with each element in mus.</td>
</tr>
<tr>
<td>MetaModel</td>
<td>List with the same length as the vector gamma. Each component of the list is a list of 3 components &quot;mu&quot;, &quot;gamma&quot; and &quot;Meta-Model&quot;:</td>
</tr>
<tr>
<td>mu</td>
<td>Scalar, the value (\mu(qmax)).</td>
</tr>
<tr>
<td>gamma</td>
<td>Positive scalar, element of the input vector gamma associated with the estimated Meta-Model.</td>
</tr>
<tr>
<td>Meta-Model</td>
<td>An RKHS Ridge Group Sparse or RKHS Group Lasso object associated with the penalty parameters (\mu) and gamma:</td>
</tr>
<tr>
<td>intercept</td>
<td>Scalar, estimated value of intercept.</td>
</tr>
<tr>
<td>teta</td>
<td>Matrix with vMax rows and (n) columns. Each row of the matrix is the estimated vector (\theta_v) for (v = 1, \ldots, \text{vMax}).</td>
</tr>
<tr>
<td>fit.v</td>
<td>Matrix with (n) rows and vMax columns. Each row of the matrix is the estimated value of (f_v = K_v \theta_v).</td>
</tr>
<tr>
<td>fitted</td>
<td>Vector of size (n), indicates the estimator of (m).</td>
</tr>
<tr>
<td>Norm.n</td>
<td>Vector of size vMax, estimated values for the Ridge penalty norm.</td>
</tr>
</tbody>
</table>
Norm.H: Vector of size vMax, estimated values of the Sparse Group penalty norm.
supp: Vector of active groups.
N supp: Vector of the names of the active groups.
SCR: Scalar, equals to $\|Y - f_0 - \sum_v K_v \theta_v\|^2$.
crit: Scalar, indicates the value of penalized criteria.
gamma.v: Vector, coefficients of the Ridge penalty norm, $\sqrt{n\gamma} \times \text{gama_v}$.
mu.v: Vector, coefficients of the Group Sparse penalty norm, $n\mu \times \text{mu_v}$.
iter: List of two components: maxIter, and the number of iterations until the convergence is achieved.
convergence: TRUE or FALSE. Indicates whether the algorithm has converged or not.
RelDiffCrit: Scalar, value of the first convergence criteria at the last iteration, $\frac{\|\theta_{lastIter} - \theta_{lastIter-1}\|^2}{\theta_{lastIter-1}}$.
RelDiffPar: Scalar, value of the second convergence criteria at the last iteration, $\frac{\text{crit}_{lastIter} - \text{crit}_{lastIter-1}}{\text{crit}_{lastIter-1}}$.

Note
For the case $\gamma = 0$ the outputs "mu" = $\mu_g$ and "Meta-Model" is the same as the one returned by the function \texttt{rkhsgrplasso}.

Author(s)
Halaleh Kamari

References

See Also
calc_Kv, mu_max, RKHSgrplasso, pen_MetMod, grplasso_q

Examples
```r
d <- 3
n <- 50
library(lhs)
X <- maximinLHS(n, d)
c <- c(0.2, 0.6, 0.8)
F <- 1; for (a in 1:d) F <- F * (abs(4 * X[, a] - 2) * c[a]) / (1 + c[a])
epsilon <- rnorm(n, 0, 1); sigma <- 0.2
Y <- F + sigma * epsilon
Dmax <- 3
kernel <- "matern"
gamma <- c(.5, .01, .001, 0)
Num <- 10
rat <- 100
```
SI_emp <- function(res, ErrPred) {
  if (is.null(ErrPred)) {
    for (i in 1:length(res)) {
      print(res[[i]]$mu)
      print(res[[i]]$gamma)
      print(res[[i]]$nsupp)
    }
  } else {
    for (i in 1:length(res)) {
      print(res[[i]]$mu)
      print(res[[i]]$gamma)
      print(res[[i]]$nsupp)
    }
  }
}

Description

Calculates the empirical sensitivity indices.

Usage

SI_emp(res, ErrPred)

Arguments

res, List, includes a sequence of estimated meta models, the solutions of the RKHS Ridge Group Sparse or RKHS Group Lasso problems. It should have the same format as the output of one of the functions: pen_MetMod, RKHSMetMod or RKHSMetMod_qmax.

ErrPred, Matrix or NULL. If matrix, each element of the matrix is the obtained prediction error associated with one RKHS meta model in "res". It should have the same format as the output of the function PredErr. Set as "NULL" by default.

Details

Details.

Value

If input ErrPred≠"NULL", Vector of the empirical sensitivity indices for the meta model with the minimum Prediction error is returned. If ErrPred="NULL", a list of the vectors is returned. Each vector is the obtained sensitivity indices associated with one meta model in "res".

Note

Note.

Author(s)

Halaleh Kamari
References


See Also

PredErr, pen_MetMod, RKHSMetMod, RKHSMetMod_qmax

Examples

d <- 3
n <- 50; nT <- 50
library(lhs)
X <- maximinLHS(n, d); XT <- maximinLHS(nT, d)
c <- c(0.2, 0.6, 0.8)
F <- 1; for (a in 1:d) F <- F*(abs(4*X[,a]-2)+c[a])/(1+c[a])
FT <- 1; for (a in 1:d) FT <- FT*(abs(4*XT[,a]-2)+c[a])/(1+c[a])
sigma <- 0.2
epsilon <- rnorm(n, 0, 1); Y <- F + sigma*epsilon
epsilonionT <- rnorm(nT, 0, 1); YT <- FT + sigma*epsilonionT
Dmax <- 3
kernel <- "matern"
fr <- c(10)
gamma = c(0.5, 0.01, 0.001)
res <- RKHSMetMod(Y, X, kernel, Dmax, gamma, fr, FALSE)
mu <- vector()
l <- length(gamma)
for(i in 1:length(fr)) mu[i] = res[[i-1]*l+1]$mu
error <- PredErr(X, XT, Y, mu, gamma, res, kernel, Dmax)
SI.minErr <- SI_emp(res, error)
SI.minErr
SI <- SI_emp(res, NULL)
SI
Index

*Topic Active groups
  grplasso_q, 5
  RKHSMetaMod_qmax, 17

*Topic Empirical
  SI_emp, 20

*Topic Gaussian Regression model
  RKHSMetaMod-package, 2

*Topic Hoeffding decomposition
  RKHSMetaMod-package, 2

*Topic Meta Model
  grplasso_q, 5
  pen_MetMod, 8
  RKHSGrplasso, 13
  RKHSMetaMod-package, 2
  RKHSMetaMod, 15
  RKHSMetaMod_qmax, 17

*Topic Optimization problem
  grplasso_q, 5
  pen_MetMod, 8
  RKHSGrplasso, 13
  RKHSMetaMod-package, 2
  RKHSMetaMod, 15
  RKHSMetaMod_qmax, 17

*Topic Package
  RKHSMetaMod-package, 2

*Topic Penalty parameter
  grplasso_q, 5
  mu_max, 7
  RKHSMetaMod_qmax, 17

*Topic Positive definite matrix
  calc_Kv, 3

*Topic Prediction error
  PredErr, 11
  RKHSMetaMod-package, 2

*Topic RKHS Group Lasso
  grplasso_q, 5
  mu_max, 7
  RKHSGrplasso, 13
  RKHSMetaMod-package, 2

  RKHSMetaMod, 15
  RKHSMetaMod_qmax, 17

*Topic RKHS Ridge Group Sparse
  pen_MetMod, 8
  RKHSMetaMod-package, 2
  RKHSMetaMod, 15
  RKHSMetaMod_qmax, 17

*Topic Sensitivity indices
  RKHSMetaMod-package, 2
  SI_emp, 20

*Topic Testing data set
  PredErr, 11
  calc_Kv, 3, 5–15, 17–19
  grplasso_q, 5, 19
  mu_max, 5, 6, 7, 15, 17–19
  pen_MetMod, 8, 11, 12, 17, 19–21
  PredErr, 11, 20, 21
  RKHSGrplasso, 9, 10, 13, 16, 17, 19
  RKHSMetaMod, 3, 4
  RKHSMetaMod (RKHSMetaMod-package), 2
  RKHSMetaMod-package, 2
  RKHSMetaMod, 11, 12, 15, 20, 21
  RKHSMetaMod_qmax, 11, 12, 17, 20, 21
  SI_emp, 20