Package ‘FGSPCA’

June 4, 2021

Title Feature Grouping and Sparse Principal Component Analysis (FGSPCA)

Version 0.1.0

Author Author1; Author2

Maintainer Author1 <anonymous@example.com>

Description
Sparse principal component analysis (SPCA) attempts to find sparse weight vectors (loadings), i.e., a loading vector with only a few 'active' (nonzero) values. The SPCA approach provides better interpretability for the principal components in high-dimensional data settings. Because the principal components are formed as a linear combination of only a few of the original variables. This package provides a modified sparse principal component analysis, Feature Grouping and Sparse Principal Component Analysis (FGSPCA), which considers additional structure information among loadings (feature grouping) as well as the sparsity (feature selection) property among loadings.

License GPL (>= 3)

Encoding UTF-8

Suggests elasticnet (>= 1.3),
sparsepca (>= 0.1.2)

Imports lars (>= 1.2),
mvtnorm (>= 1.1),
Matrix (>= 1.2),
glmnet (>= 4.1)

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

R topics documented:

`convcheck` *Convergence check of two successive matrices*

Description

Check the convergence of two successive matrices

Usage

```
convcheck(beta1, beta2)
```

Arguments

`beta1`	B_1 a matrix
`beta2`	B_2 a matrix

Details

The two matrices B_1 and B_2 may be different only in their column signs with respective to each column.

Value

the largest absolute value of the difference of their corresponding columns

---

`coordinate_descent_enet` *The coordinate descent for the standard elastic net regression problem*

Description

The objective function of the standard elastic net regression problem is

$$1/(2n)\|Y - X\beta\|_2^2 + \lambda(\alpha\|\beta\|_1 + (1-\alpha)/2\|\beta\|_2^2).$$

It is the standard version of the elastic net problem.

Usage

```
coordinate_descent_enet(
  x,
  y,
  lambda,
  alpha,
  max.steps = 100,
  condition_tol = 0.001,
  loss_return = FALSE
)
```

Arguments

`x`	the data matrix $X_{n\times p}$
`y`	the response vector $Y_{n\times 1}$
`lambda`	the λ in the elastic net loss function
`alpha`	the α in the elastic net loss function
`max.steps`	maximum steps, the maximum number of steps for the updating, `100` (default)
`condition_tol`	the tolerance for the condition to stop, `1e-3` (default)
`loss_return`	whether to return loss or not, `FALSE` (default)

Value

β the solution β or a list

a list of β, a sequence of the objective values during the CD process, a sequence of the loss values, a sequence of the penalty values, `list(beta=beta,obj=obj,loss=loss,pen=pen)`.

`coordinate_descent_LASSO`

The coordinate descent for LASSO (elastic net) regression

Description

The coordinate descent to solve the LASSO (elastic net) regression problem. It is not the standard version of the elastic net problem, so we use `coordinate_descent_LASSO`. The standard version of the elastic net problem is termed `coordinate_descent_enet`.

Usage

```
coordinate_descent_LASSO(
  x,
  y,
  paras,
  max.steps = 100,
  condition_tol = 0.001,
  loss_return = FALSE
)
```

Arguments

- `x` the data matrix $X_{n \times p}$
- `y` the response vector $Y_{n \times 1}$
- `paras` the combination of parameters λ_2, λ_1
- `max.steps` maximum steps, the maximum number of steps for the updating, `100` (default)
- `condition_tol` the tolerance for the condition to stop, `1e-3` (default)
- `loss_return` whether to return loss or not, `FALSE` (default)

Details

The objective function of the lasso (elastic net) is

$$1/(2n)\|Y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\beta\|_2^2.$$

While the soft thresholding method applied to the matrix version may not lead to convergence, the updating using the soft thresholding along each coordinate can guarantee the convergence.

Value

β the solution β or a list

a list of β, a sequence of the objective values during the CD process, a sequence of the loss values, a sequence of the penalty values, `list(beta=beta,obj=obj,loss=loss,pen=pen)`.

`different_beta_solver` *Different beta solvers*

Description

Different beta solvers

Usage

```
different_beta_solver(
  x,
  y,
  lambda2,
  lambda1,
  sparse,
  solver_type = "spcasolver"
)
```

Arguments

- `x` the data matrix $X_{n \times p}$
- `y` the response vector $Y_{n \times 1}$
- `lambda2` λ_2 in the elastic loss function
- `lambda1` λ_1 in the elastic loss function
- `sparse` `sparse = c("penalty","varnum")`
- `solver_type` `solver_type=c("spcasolver","lassocd","glmnet")`

Value

the solution β

`enet_loss` *The elastic net loss function*

Description

The elastic net loss function (not the standard version)

Usage

```
enet_loss(x, y, beta, lambda1, lambda2, verbose = FALSE)
```

Arguments

`x`	the data matrix $X_{n \times p}$
`y`	the response vector $Y_{n \times 1}$
`beta`	β the estimation of β
`lambda1`	the λ_1 in the loss function
`lambda2`	the λ_2 in the loss function
`verbose`	whether to return a list, `FALSE` (default)

Details

The standard version of the elastic net problem is defined as follows,

$$1/(2n)\|Y - X\beta\|_2^2 + \lambda(\alpha\|\beta\|_1 + (1-\alpha)/2\|\beta\|_2^2).$$

The objective function of the elastic net we use in this function is defined as

$$1/(2n)\|Y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\beta\|_2^2.$$

It is not the standard version of the elastic net problem.

Value

the value of the loss function or a list `c(loss,pen,obj)`

FGSPCA *Feature Grouping and Sparse PCA with L2 loss function.*

Description

Feature Grouping and Sparse PCA with L2 loss function.

Usage

```
FGSPCA(
  x,
  B_init,
  K,
  para,
  type = "predictor",
  use.corr = FALSE,
  max.iter = 200,
  trace = TRUE,
  eps.conv = 0.001,
  tau_S = 0.05,
  lambda2 = 0.1,
  lambda3 = 0.1,
  v = 1,
  c = 1.02,
  iter_m_max = 100,
  iter_k_max = 50,
  condition_tol = 1e-05,
  nnConstraint = FALSE,
  sparseTruncated = TRUE
)
```

Arguments

`x`	the data matrix X
`B_init`	the initial value of matrix B
`K`	the number of principal components
`para`	a list of λ_1 with its length being `K`
`type`	type of X, which can take values of `type=c("predictor","Gram")`. If `type="Gram"` the model should deal with the root matrix of X. If `type="predictor"` the model should directly deal with the matrix X.
`use.corr`	`FALSE` (default). When `type="predictor"` we need to scale the data, `scale=use.corr` .
`max.iter`	`200` (default) the max iteration of the `A-B` updating procedure
`trace`	`TRUE` (default) whether to do the print or not (print the number of the current iteration and the error at the iteration)
`eps.conv`	$1e-3$ (default) the convergence criterion for the `A-B` updating procedure
`tau_S`	τ, a global τ, which is assigned to $\tau_1 = \tau_2 = \tau$.
`lambda2`	λ_2, the tuning parameter corresponding to $p_2(\cdot)$

`lambda3` λ_3, the tuning parameter corresponding to $p_3(\cdot)$

`v` the initial value of the Lagrange multiplier, with default `v=1.0`.

`c` the acceleration constant to speed up the convergence procedure, default `c=1.02`.

`iter_m_max` the maximum number of the outer iterations (i.e. the m-iteration), default `iter_m_max=100`

`iter_k_max` the maximum number of the inner iterations (i.e. the k-iteration), default `iter_k_max=50`

`condition_tol` the conditional tolerance of both the outer and inner iterations, default `condition_tol=1e-5`

`nnConstraint` Boolean, indicating the non-negative constraint is true or false, default `FALSE`

`sparseTruncated` Boolean, indicating whether to use the truncated L1 penalty or not for sparsity, default `TRUE`

Details

The main function of the FGSPCA problem with L2 loss using the alternating updating, which updates A while fixing B and updates B while fixing A.

Value

pev Percentage of Explained Variance

var.all the total variance

loadings the final normalized loadings of B

Alpha the final eqnA from the last update

ab_errors the `A-B` updating errors

type (same as the input) the type of matrix X

K (same as the input) the number of principal components

para (same as the input) a list of λ_1 with its length being `K`

lambda2 (same as the input) λ_2

lambda3 (same as the input) λ_3

vn variable names

Mloss a matrix of loss with shape of $i \times K$, where i is the number of iterations.

Gloss global loss

Sloss sum loss

Tloss tail loss

LossList with each element being a list of the objective values of each subproblem. $K * (i - 1) + j$
.

Examples

```
library(elasticnet)  # to use the data "pitprop"
library(sparsepca)
data(pitprops)
# ## elasticnet::spca is the Sparse PCA of Zou (2006).
K <- 6
para <- c(0.06, 0.16, 0.1, 0.5, 0.5, 0.5)
out1 <- elasticnet::spca(
  pitprops, K=6, type="Gram", sparse="penalty", trace=TRUE,
  para=c(0.06,0.16,0.1,0.5,0.5,0.5))
```

```
K <- k <- 6
X <- rootmatrix(pitprops)
rspca.results <- sparsepca::rspca(X, k, alpha=1e-3, beta=1e-3, center=TRUE, scale=FALSE)

K <- 6
B_init <- rspca.results$loadings
tau_S=0.05; lambda1=0.01; lambda2=0.01; lambda3=0.1
para <- rep(lambda1, K)

out <- FGSPCA(
  pitprops, B_init, K, para, type="Gram",
  tau_S=tau_S, lambda2=lambda2, lambda3=lambda3)
NB <- out$loadings  # the normalized loadings of FGSPCA

(fgspca.pev <- round(out$pev *100, 3))
(fgspca.cpev <- round(cumsum(out$pev) * 100, 3))
```

FSGFunL2 *The Feature Selection and Grouping Function FSGFun with L2 loss objective.*

Description

The FSGFun with L2 loss is a subproblem of the FGSPCA with L2 loss.

Usage

```
FSGFunL2(
  x,
  y,
  beta,
  tau_S,
  lambda1,
  lambda2,
  lambda3,
  v = 1,
  c = 1.02,
  iter_m_max = 100,
  iter_k_max = 50,
  condition_tol = 1e-05,
  nnConstraint = FALSE,
  sparseTruncated = TRUE,
  loss_return = FALSE
)
```

Arguments

x	the data matrix $X_{n \times p}$, where n is the number of observations, p is the number of features.
y	the response vector $Y_{n \times 1}$ with length n

beta	β, the estimation of β
tau_S	τ, a global τ, which is assigned to $\tau_1 = \tau_2 = \tau$.
lambda1	λ_1, the tuning parameter corresponding to $p_1(\cdot)$
lambda2	λ_2, the tuning parameter corresponding to $p_2(\cdot)$
lambda3	λ_3, the tuning parameter corresponding to $p_3(\cdot)$
v	the initial value of the Lagrange multiplier, with default `v=1.0`.
c	the acceleration constant to speed up the convergence procedure, default `c=1.02`.
iter_m_max	the maximum number of the outer iterations (i.e. the m-iteration), default `iter_m_max=100`
iter_k_max	the maximum number of the inner iterations (i.e. the k-iteration), default `iter_k_max=50`
condition_tol	the conditional tolerance of both the outer and inner iterations, default `condition_tol=1e-5`
nnConstraint	Boolean, indicating the non-negative constraint is true or false, default `FALSE`
sparseTruncated	Boolean, indicating whether to use the truncated L1 penalty or not for sparsity, default `TRUE`
loss_return	Boolean, indicating whether return the loss or not, default `FALSE`

Details

The parameters of `x`, `y`, `beta`, `lambda1`, `lambda2`, `lambda3` are in inherit from `ObjFunL2`.

Value

the solution β or with the corresponding loss if `loss_return=TRUE`

`ObjFunL2` *The objective function using L2 loss with three penalty functions*

Description

The objective function using L2-loss with three penalty functions. An extension of the elastic net regression.

Usage

```
ObjFunL2(
  x,
  y,
  tau_S1,
  tau_S2,
  lambda1,
  lambda2,
  lambda3,
  beta,
  Bjj,
  SF,
  SFc,
  SE,
  SEc,
  SN
)
```

Arguments

x — the data matrix $X_{n\times p}$, where n is the number of observations, p is the number of features.

y — the response vector $Y_{n\times 1}$ with length n

tau_S1 — τ_1, the controlling parameter corresponding to $p_1(\cdot)$, which determines when the small values of $|\beta_j|$ will be penalized.

tau_S2 — τ_2, the controlling parameter corresponding to $p_2(\cdot)$, which determines when the small difference values of $|\beta_j - \beta_{j'}|$ will be penalized.

lambda1 — λ_1, the tuning parameter corresponding to $p_1(\cdot)$

lambda2 — λ_2, the tuning parameter corresponding to $p_2(\cdot)$

lambda3 — λ_3, the tuning parameter corresponding to $p_3(\cdot)$

beta — β, the estimation of β

Bjj — a matrix of $p \times p$ with element $\beta_{jj'} = \beta_j - \beta_{j'}$.

SF — the $\mathcal{F}$ set, p-length vector of indicator 0-1. Its value is 1 if $|\beta_j| \le \tau_1$. Otherwise 0.

SFc — the $\mathcal{F}^c$ set, p-length vector of indicator 0-1. Its value is 1 if $|\beta_j| > \tau_1$. Otherwise 0.

SE — the $\mathcal{E}$ set, a matrix of $p \times p$ with indicator 0-1. Its value is 1 if $|\beta_j - \beta_{j'}| \le \tau_2$. Otherwise 0. Note if $\tau_2 = 0, j = j'$, then $0 <= 0$ is true, the diagonal of $\mathcal{E}$ is 1. We should set `SEc <-(1-SE)`, `SE <-SE -diag(p)` after the calculation of $\mathcal{E}$.

SEc — the $\mathcal{E}^c$ set, a matrix of $p \times p$ with indicator 0-1.

SN — the $\mathcal{N}$ set, p-length vector of indicator 0-1. Its value is 1 when $\beta_j < 0$, corresponding to $\min(\beta_j, 0)$.

Details

The objective function using L2 loss is defined as follows,

$$\frac{1}{2n}\|Y - X\beta\|^2 + \lambda_1 p_1(\beta) + \lambda_2 p_2(\beta) + \lambda_3 p_3(\beta).$$

The three penalties are as follows,

$$p_1(\beta) = \sum_{j=1}^{p} \min\{\frac{|\beta_j|}{\tau_1}, 1\},$$

$$p_2(\beta) = \sum_{j<j',(j,j')\in E} \min\{\frac{|\beta_j - \beta_{j'}|}{\tau_2}, 1\},$$

$$p_3(\beta) = \sum_{j=1}^{p} (\min\{\beta_j, 0\})^2.$$

Value

the value of the objective function

ObjFun_FG_S_PCA_L2 *Objective Function of Feature Grouping and Sparse PCA with L2 loss.*

Description

Objective Function of Feature Grouping and Sparse PCA with L2 loss.

Usage

```
ObjFun_FG_S_PCA_L2(
  x,
  A,
  B,
  tau_S,
  lambda1,
  lambda2 = 0.1,
  lambda3 = 0.1,
  nnConstraint = FALSE,
  sparseTruncated = TRUE
)
```

Arguments

x	the data matrix $X_{n \times p}$, where n is the number of observations, p is the number of features.
A	matrix A in the FGSPCA problem
B	matrix B in the FGSPCA problem
tau_S	τ, a global τ, which is assigned to $\tau_1 = \tau_2 = \tau$.
lambda1	λ_1, the tuning parameter corresponding to $p_1(\cdot)$
lambda2	λ_2, the tuning parameter corresponding to $p_2(\cdot)$
lambda3	λ_3, the tuning parameter corresponding to $p_3(\cdot)$
nnConstraint	Boolean, indicating the non-negative constraint is true or false, default FALSE
sparseTruncated	Boolean, indicating whether to use the truncated L1 penalty or not for sparsity, default TRUE

Details

The L2 loss is defined as follows

$$(X - XBA^T)^2.$$

The penalty function with three parts is defined as follows,

$$\Psi(\beta) = \lambda_1 p_1(\beta) + \lambda_2 p_2(\beta) + \lambda_3 p_3(\beta)$$

where

$$p_1(\beta) = \sum_{j=1}^{p} \min\{\frac{|\beta_j|}{\tau_1}, 1\},$$

$$p_2(\beta) = \sum_{j<j',(j,j')\in E} \min\{\frac{|\beta_j - \beta_{j'}|}{\tau_2}, 1\},$$

$$p_3(\beta) = \sum_{j=1}^{p} (\min\{\beta_j, 0\})^2.$$

With the setting of lambda2=0, nnConstraint=FALSE, sparseTruncated=FALSE, it corresponds to the same objective function of sparse PCA; \
ObjFun_FG_S_PCA_L2(x, A, B, tau_S, lambda1, lambda2=0, lambda3=0.1, nnConstraint=FALSE, sparseTruncated=FALSE

With setting of lambda2=0, nnConstraint=FALSE, sparseTruncated=TRUE, corresponds to Objective function of truncated sparse PCA. \
ObjFun_FG_S_PCA_L2(x, A, B, tau_S, lambda1, lambda2=0, lambda3=0.1, nnConstraint=FALSE, sparseTruncated=TRUE

By tuning the different setting of `lambda2`, `nnConstraint`, `sparseTruncated`, we get different combination of models.

`PenaltyFun` *The penalty function consisting of three parts.*

Description

The penalty function consisting of three parts.

Usage

```
PenaltyFun(
  beta,
  tau_S,
  lambda1,
  lambda2,
  lambda3,
  nnConstraint = FALSE,
  sparseTruncated = TRUE
)
```

Arguments

`beta`	β, the estimation of β
`tau_S`	τ a global τ, which is assigned to $\tau_1 = \tau_2 = \tau$.
`lambda1`	λ_1, the tuning parameter corresponding to $p_1(\cdot)$
`lambda2`	λ_2, the tuning parameter corresponding to $p_2(\cdot)$
`lambda3`	λ_3, the tuning parameter corresponding to $p_3(\cdot)$
`nnConstraint`	Boolean, indicating the non-negative constraint is true or false, default `FALSE`
`sparseTruncated`	Boolean, indicating whether use the truncated L1 penalty or not for sparsity, default `TRUE`

Details

The three penalties are as follows,

$$p_1(\beta) = \sum_{j=1}^{p} \min\{\frac{|\beta_j|}{\tau_1}, 1\},$$

$$p_2(\beta) = \sum_{j<j',(j,j')\in E} \min\{\frac{|\beta_j - \beta_{j'}|}{\tau_2}, 1\},$$

$$p_3(\beta) = \sum_{j=1}^{p} (\min\{\beta_j, 0\})^2.$$

`rootmatrix` *Calculate the root matrix of the given square matrix using eigendecomposition*

Description

Calculate the root matrix of the given square matrix using eigendecomposition

Usage

```
rootmatrix(x)
```

Arguments

x	the given square matrix with size $n \times n$

Value

the root matrix

`solvebeta` *The SPCA solver of beta for the elastic net problem*

Description

The beta solver using the sparse PCA.

Usage

```
solvebeta(
  x,
  y,
  paras,
  max.steps,
  sparse = "penalty",
  eps = .Machine$double.eps
)
```

Arguments

x	the data matrix $X_{n \times p}$
y	the response vector $Y_{n \times 1}$
paras	the combination of parameters (λ_2, λ_1)\ lambda = paras[1] (which is lambda2) and lambda1 = paras[2] .
max.steps	100 (default) the maximum number of steps for the updating
sparse	sparse = c("penalty","varnum")
eps	the tolerance of the stopping criterion for the termination

Details

The standard objective function of elastic net is

$$1/(2n)\|Y - X\beta\|_2^2 + \lambda(\alpha\|\beta\|_1 + (1-\alpha)/2\|\beta\|_2^2).$$

But here we use the following objective function

$$1/(2n)\|Y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2/2\|\beta\|_2^2.$$

Value

the solution β in each subproblem

References

- Zou, Hui, Trevor Hastie, and Robert Tibshirani. "Sparse principal component analysis." Journal of computational and graphical statistics 15.2 (2006): 265-286.

---

total_variance_explained

Adjusted Total Variance by Zou. et.al.(2006)

Description

To calculate the Total Variance Explained by the new normalized loadings NB. The loadings of the ordinary principal components are uncorrelated and orthogonal, where the orthogonal property, and the uncorrelated property are satisfied together.

Usage

```
total_variance_explained(x, NB)
```

Arguments

x	the data matrix by $n \times p$. It should not be the sample covariance matrix, nor the correlation matrix.
NB	normalized matrix of loading vectors

Value

a list of c(pev_svd,pev_tr,pev_spca,cum_svd,cum_tr,cum_spca).

References

- Zou, Hui, Trevor Hastie, and Robert Tibshirani. "Sparse principal component analysis." Journal of computational and graphical statistics 15.2 (2006): 265-286.

Index