Package 'comlasso'

February 10, 2019

Type Package

Title Primal path algorithm for compositional data (comlasso)

Version 0.2

Date 2019-02-09

Author Jong-Jun Jeon, Yondai Kim, Hyowon An, Sungho Won and Hosik Choi

Maintainer Hosik Choi <choi.hosik@gmail.com>

Depneds R (>=3.2.2), Rglpk (>=0.6-2)

Description This package provides functions for fitting the entire solution path algorithm for compositional data analysis

License GPL(>=2)

First version Sep/19/2016

R topics documented:

comlasso *Primal path algorithm for compositional data*

Description

Fit the entire solution path of lasso under zero sum constraint for regression & classification problem

Usage

```
comlasso(ltype = c("regression", "classification"), n, p, K, X.raw, y,
  gam = 1e2, weights = NULL, max.steps = length(K)*min(n,sum(K))+1, lam.min = 0,
  tol = 1e-08, trace = FALSE)
```

Arguments

`ltype` choose either one among c("regression", "classification")

`n` sample size

`p` number of components

`K` vector of the number of variables belonging to each component

`X.raw` n times K[1]+K[2]+...+K[p] matrix

`y` response vector with length n

`gam` Knot value for Huber loss function. For classification, default value of gam is 1e2. For classification problem, gam should be <= 1.

`weights` For adaptive lasso penalty,

`max.steps` steps: default value is min(n, K) + 1

`lam.min` mininum value of lambda. Default value is 0

`tol` relative zero

`trace` To check status of solution

Details

See example codes

Value

return classo object and see predict.classo

Note

n<p & blockwise zero sum for several components

Author(s)

Jong-Jun Jeon, Yondai Kim, Hyowon An, Sungho Won and Hosik Choi (2016) Primal path algorithm for compositional data analysis

References

Primal path algorithm for compositional data analysis

See Also

predict.class, stability_classo

Examples

```
data(sand)
mdat <- sand[,-1]
x <- as.matrix(mdat[,1:3])/100
x <- log(x)
y <- log(mdat[,4])

n <- 39; p <- 3
# lasso
```

```
fit1 <- comlasso("regression", n=n, p=1, K=p, X.raw=x, y=y, weights=c(1,1,1))
w <- 1/(abs(fit1$beta[[1]])/sum(abs(fit1$beta[[1]])))
fit2 <- comlasso("regression", n=n, p=1, K=p, X.raw=x, y=y, weights=w)

par(mfrow=c(1,2))
matplot(t(fit1$beta.rec[[1]]),type="b",xaxt="n",ylab=expression(hat(beta)[j](lambda)),
  ylim=c(-0.6,0.6), main="lasso")
axis(1,at=c(1,2,3))
abline(h=0,lty=2,lwd=0.7)
matplot(t(fit2$beta.rec[[1]]),type="b",xaxt="n",ylab=expression(hat(beta)[j](lambda)),
  ylim=c(-0.6,0.6), main="alasso")
axis(1,at=c(1,2,3))
abline(h=0,lty=2,lwd=0.7)
```

predict.comlasso *Make predictions or extract coefficients from a fitted comlasso model*

Description

While comlasso() produces the entire path of solutions, predict.comlasso allows one to extract a prediction at a particular point along the path.

Usage

```
## S3 method for class 'comlasso'
predict(object, newx, s, type=c("fit","coefficients"),
  mode=c("step","fraction","norm","lambda"), ...)
```

Arguments

object	comlasso object
newx	If type="fit", then newx should be the x values at which the fit is required. If type="coefficients", then newx can be omitted.
s	a value, or vector of values, indexing the path. Its values depends on the mode= argument. By default (mode="step"), s should take on values between 0 and p (e.g., a step of 1.3 means .3 of the way between step 1 and 2.)
type	If type="fit", predict returns the fitted values. If type="coefficients", predict returns the coefficients. Abbreviations allowed. mode
mode	mode="step" means the s= argument indexes the comlasso step number, and the coefficients will be returned corresponding to the values corresponding to step s. If mode="fraction", then s should be a number between 0 and 1, and it refers to the ratio of the L1 norm of the coefficient vector, relative to the norm at the full LS solution. mode="norm" means s refers to the L1 norm of the coefficient vector. mode="lambda" uses the lasso regularization parameter for s; for other models it is the maximal correlation (does not make sense for comlasso models).
...	additional parameters

Details

comlasso is described in detail in Jeon, Kim, An, Won and Choi (2016). It computes the complete lasso solution simultaneously for ALL values of the shrinkage parameter in the same computational cost as a least squares fit without penalization.

Value

Either a vector/matrix of fitted values, or a vector/matrix of coefficients.

Note

predict / predict.comlasso

Author(s)

Jong-Jun Jeon, Yondai Kim, Hyowon An, Sungho Won and Hosik Choi (2016) Primal path algorithm for compositional data analysis

References

Jeon, Kim, An, Won and Choi (2016) "Primal path algorithm for compositional data analysis"

See Also

stability_comlasso

Examples

```
data(sand)
mdat <- sand[,-1]
x <- as.matrix(mdat[,1:3])/100
x <- log(x)
y <- log(mdat[,4])

n <- 39; p <- 3
# lasso
fit <- comlasso("regression", n=n, p=1, K=p, X.raw=x, y=y, weights=c(1,1,1))
predict(fit, x, s=c(fit$lambda,0.5), type="fit",mode="lambda")
predict(fit, x, s=c(0,0.5,1), type="fit",mode="fraction")
predict(fit, x, s=1:2, type="fit",mode="step")
predict(fit, x, s=0.5, type="fit",mode="norm")
```

Index