Package 'OHPL'

August 8, 2017

Type Package

Title Ordered Homogeneity Pursuit Lasso for Group Variable Selection

Version 1.3

Maintainer Nan Xiao <me@nanx.me>

Description Ordered homogeneity pursuit lasso (OHPL) algorithm for group variable selection proposed in Lin et al. (2017) <DOI:10.1016/j.chemolab.2017.07.004>. The OHPL method exploits the homogeneity structure in high-dimensional data and enjoys the grouping effect to select groups of important variables automatically. This feature makes it particularly useful for high-dimensional datasets with strongly correlated variables, such as spectroscopic data.

License GPL-3 | file LICENSE

URL https://ohpl.io, https://github.com/road2stat/OHPL

Encoding UTF-8

LazyData true

BugReports https://github.com/road2stat/OHPL/issues

Depends R (>= 3.0.2)

Imports glmnet, pls, mvtnorm

RoxygenNote 5.0.1

NeedsCompilation no

Author You-Wu Lin [aut], Nan Xiao [cre]

Repository CRAN

Date/Publication 2017-08-08 17:18:59 UTC

R topics documented:

`OHPL-package` *Ordered Homogeneity Pursuit Lasso for Group Variable Selection*

Description

OHPL implements the ordered homogeneity pursuit lasso (OHPL) algorithm for group variable selection proposed in Lin et al. (2017) <DOI:10.1016/j.chemolab.2017.07.004>. The OHPL method takes the homogeneity structure in high-dimensional data into account and enjoys the grouping effect to select groups of important variables automatically. This feature makes it particularly useful for high-dimensional datasets with strongly correlated variables, such as spectroscopic data.

Details

Package:	OHPL
Type:	Package
License:	GPL-3

References

You-Wu Lin, Nan Xiao, Li-Li Wang, Chuan-Quan Li, and Qing-Song Xu (2017). Ordered homogeneity pursuit lasso for group variable selection with applications to spectroscopic data. *Chemometrics and Intelligent Laboratory Systems* 168, 62-71. https://doi.org/10.1016/j.chemolab.2017.07.004

`beer` *The beer dataset*

Description

The beer dataset contains 60 samples published by Norgaard et al. Recorded with a 30mm quartz cell on the undiluted degassed beer and measured from 1100 to 2250 nm (576 data points) in steps of 2 nm.

Usage

```
data(beer)
```

References

Norgaard, L., Saudland, A., Wagner, J., Nielsen, J. P., Munck, L., & Engelsen, S. B. (2000). Interval partial least-squares regression (iPLS): a comparative chemometric study with an example from near-infrared spectroscopy. *Applied Spectroscopy*, 54(3), 413–419.

Examples

```
data("beer")
x.cal = beer$xtrain
dim(x.cal)
x.test = beer$xtest
dim(x.test)
y.cal = beer$ytrain
dim(y.cal)
y.test = beer$ytest
dim(y.test)

X = rbind(x.cal, x.test)
y = rbind(y.cal, y.test)
n = nrow(y)

set.seed(1001)
samp.idx = sample(1L:n, round(n * 0.7))
X.cal = X[samp.idx, ]
y.cal = y[samp.idx]
X.test = X[-samp.idx, ]
y.test = y[-samp.idx]
```

---

cv.OHPL *Cross-Validation for Ordered Homogeneity Pursuit Lasso*

Description

This function uses cross-validation to help select the optimal number of variable groups and the value of gamma.

Usage

```
cv.OHPL(X.cal, y.cal, maxcomp, gamma = seq(0.1, 0.9, 0.1), X.test, y.test,
  cv.folds = 5L, G = 30L, type = c("max", "median"), scale = TRUE,
  pls.method = "simpls")
```

Arguments

X.cal	Predictor matrix (training)
y.cal	Response matrix with one column (training)
maxcomp	Maximum number of components for PLS
gamma	A vector of the gamma sequence between (0, 1).
X.test	X.test Predictor matrix (test)
y.test	y.test Response matrix with one column (test)
cv.folds	Number of cross-validation folds
G	Maximum number of variable groups
type	Find the maximum absolute correlation ("max") or find the median of absolute correlation ("median"). Default is "max".
scale	Should the predictor matrix be scaled? Default is TRUE.
pls.method	Method for fitting the PLS model. Default is "simpls". See the details section in plsr for all possible options.

Value

A list containing the optimal model, RMSEP, Q2, and other evaluation metrics. Also the optimal number of groups to use in group lasso.

Examples

```
data("wheat")

X = wheat$x
y = wheat$protein
n = nrow(wheat$x)

set.seed(1001)
samp.idx = sample(1L:n, round(n * 0.7))
X.cal = X[samp.idx, ]
y.cal = y[samp.idx]
X.test = X[-samp.idx, ]
y.test = y[-samp.idx]

# This needs to run for a while
## Not run:
cv.fit = cv.OHPL(
  x, y, maxcomp = 6, gamma = seq(0.1, 0.9, 0.1),
  x.test, y.test, cv.folds = 5, G = 30, type = "max")
# the optimal G and gamma
```

```
cv.fit$opt.G
cv.fit$opt.gamma
## End(Not run)
```

`dlc` *Compute D, L, and C in Fisher Optimal Partitions Algorithm*

Description

Compute D, L, and C in Fisher Optimal Partitions Algorithm

Usage

```
dlc(X, maxk)
```

Arguments

`X`	a set of samples
`maxk`	maximum number of k

Value

The D, L, and C statistics in Fisher optimal partitions algorithm

---

`FOP` *Fisher Optimal Partition*

Description

The Fisher optimal partition algorithm.

Usage

```
FOP(X, k, C)
```

Arguments

`X`	a set of samples
`k`	number of classes
`C`	statistic from the output of `dlc`

Value

index vector for each sample's classification

References

W. D. Fisher (1958). On grouping for maximum homogeneity. *Journal of the American Statistical Association*, vol. 53, pp. 789–798.

Examples

```
X = matrix(c(
  9.3, 1.8, 1.9, 1.7, 1.5, 1.3,
  1.4, 2.0, 1.9, 2.3, 2.1))
C = dlc(X, maxk = 8)$C
F = FOP(X, 8, C)
```

OHPL *Ordered Homogeneity Pursuit Lasso*

Description

This function fits the ordered homogeneity pursuit lasso (OHPL) model.

Usage

```
OHPL(x, y, maxcomp, gamma, cv.folds = 5L, G = 30L, type = c("max",
  "median"), scale = TRUE, pls.method = "simpls")
```

Arguments

x	Predictor matrix.
y	Response matrix with one column.
maxcomp	Maximum number of components for PLS.
gamma	A number between (0, 1) for generating the gamma sequence. An usual choice for gamma could be `n * 0.05`, where `n` is a number in 2, 3, ..., 19.
cv.folds	Number of cross-validation folds.
G	Maximum number of variable groups.
type	Find the maximum absolute correlation (`"max"`) or find the median of absolute correlation (`"median"`). Default is `"max"`.
scale	Should the predictor matrix be scaled? Default is `TRUE`.
pls.method	Method for fitting the PLS model. Default is `"simpls"`. See section "Details" in `plsr` for all possible options.

Value

A list of fitted OHPL model object with performance metrics.

References

You-Wu Lin, Nan Xiao, Li-Li Wang, Chuan-Quan Li, and Qing-Song Xu (2017). Ordered homogeneity pursuit lasso for group variable selection with applications to spectroscopic data. *Chemometrics and Intelligent Laboratory Systems* 168, 62-71. https://doi.org/10.1016/j.chemolab.2017.07.004

Examples

```
# generate simulation data
dat = OHPL.sim(
  n = 100, p = 100, rho = 0.8,
  coef = rep(1, 10), snr = 3, p.train = 0.5,
  seed = 1010)

# split training and test set
x = dat$x.tr
y = dat$y.tr
x.test = dat$x.te
y.test = dat$y.te

# fit the OHPL model
fit = OHPL(x, y, maxcomp = 3, gamma = 0.5, G = 10, type = "max")
# selected variables
fit$Vsel
# make predictions
y.pred = predict(fit, x.test)
# compute evaluation metric RMSEP, Q2 and MAE for the test set
perf = OHPL.RMSEP(fit, x.test, y.test)
perf$RMSEP
perf$Q2
perf$MAE
```

OHPL.RMSEP *Compute RMSEP, MAE, and Q2 for a Test Set*

Description

This function makes predictions on new data and computes the performance evaluation metrics RMSEP, MAE, and Q2.

Usage

```
OHPL.RMSEP(object, newx, newy)
```

Arguments

object	An An object of class OHPL fitted by OHPL.
newx	Predictor matrix of the new data.
newy	Response matrix of the new data (matrix with one column).

Value

A list of the performance metrics.

Examples

```
# generate simulation data
dat = OHPL.sim(
  n = 100, p = 100, rho = 0.8,
  coef = rep(1, 10), snr = 3, p.train = 0.5,
  seed = 1010)

# split training and test set
x = dat$x.tr
y = dat$y.tr
x.test = dat$x.te
y.test = dat$y.te

# fit the OHPL model
fit = OHPL(x, y, maxcomp = 3, gamma = 0.5, G = 10, type = "max")
# compute evaluation metric RMSEP, Q2 and MAE for the test set
perf = OHPL.RMSEP(fit, x.test, y.test)
perf$RMSEP
perf$Q2
perf$MAE
```

OHPL.sim *Generate Simulation Data for Benchmarking Sparse Regressions (Gaussian Response)*

Description

Generate simulation data (Gaussian case) following the settings in Xiao and Xu (2015).

Usage

```
OHPL.sim(n = 100, p = 100, rho = 0.8, coef = rep(1, 10), snr = 3,
  p.train = 0.5, seed = 1001)
```

Arguments

n	Number of observations.
p	Number of variables.
rho	Correlation base for generating correlated variables.
coef	Vector of non-zero coefficients.
snr	Signal-to-noise ratio (SNR).
p.train	Percentage of training set.
seed	Random seed for reproducibility.

Value

List of `x.tr`, `x.te`, `y.tr`, and `y.te`.

Author(s)

Nan Xiao <https://nanx.me>

References

Nan Xiao and Qing-Song Xu. (2015). Multi-step adaptive elastic-net: reducing false positives in high-dimensional variable selection. *Journal of Statistical Computation and Simulation* 85(18), 3755–3765.

Examples

```
dat = OHPL.sim(
  n = 100, p = 100, rho = 0.8,
  coef = rep(1, 10), snr = 3, p.train = 0.5,
  seed = 1010)

dim(dat$x.tr)
dim(dat$x.te)
```

`predict.OHPL` *Make Predictions Based on the Fitted OHPL Model*

Description

Make predictions on new data by an OHPL model object.

Usage

```
## S3 method for class 'OHPL'
predict(object, newx, ncomp = NULL, type = "response", ...)
```

Arguments

`object`	An object of class `OHPL` fitted by `OHPL`.
`newx`	Predictor matrix of the new data.
`ncomp`	Optimal number of components. If is `NULL`, the optimal number of components stored in the model object will be used.
`type`	Prediction type.
`...`	Additional parameters.

Value

Numeric matrix of the predicted values.

Examples

```
# generate simulation data
dat = OHPL.sim(
  n = 100, p = 100, rho = 0.8,
  coef = rep(1, 10), snr = 3, p.train = 0.5,
  seed = 1010)

# split training and test set
x = dat$x.tr
y = dat$y.tr
x.test = dat$x.te
y.test = dat$y.te

# fit the OHPL model
fit = OHPL(x, y, maxcomp = 3, gamma = 0.5, G = 10, type = "max")
# make predictions
y.pred = predict(fit, x.test)
y.pred
```

---

proto *Extract the Prototype from Each Variable Group*

Description

This function extracts the prototype from each variable group.

Usage

```
proto(X, y, groups, type = c("max", "median"), mu = NULL)
```

Arguments

X	Predictor matrix
y	Response matrix with one column
groups	An group index vector containing the group number each variable belongs to. For example: `c(1, 1, 1, 1, 1, 2, 2, 2, ...)`. Variable groups can be generated by the Fisher optimal partition algorithm implemented in FOP.
type	The rule for extracting the prototype. Possible options are `"max"` and `"median"`.
mu	The mean value of y for standarization. Default is NULL, which uses the sample mean of y.

Value

The prototypes (variable index) extracted from each group (cluster).

soil *The soil dataset*

Description

The soil dataset contains 108 sample measurements from the wavelength range of 400–2500 nm (visible and near infrared spectrum) published by Rinnan et al.

Usage

```
data(soil)
```

References

Rinnan, R., & Rinnan, A. (2007). Application of near infrared reflectance (NIR) and fluorescence spectroscopy to analysis of microbiological and chemical properties of arctic soil. *Soil biology and Biochemistry*, 39(7), 1664–1673.

Examples

```
data("soil")

X = soil$x
y = soil$som
n = nrow(soil$x)

set.seed(1001)
samp.idx = sample(1L:n, round(n * 0.7))
X.cal = X[samp.idx, ]
y.cal = y[samp.idx]
X.test = X[-samp.idx, ]
y.test = y[-samp.idx]
```

wheat *The wheat dataset*

Description

The wheat dataset contains 100 wheat samples with specified protein and moisture content, published by J. Kalivas. Samples were measured by diffuse reflectance as log (I/R) from 1100 to 2500 nm (701 data points) in 2 nm intervals.

Usage

```
data(wheat)
```

References

Kalivas, J. H. (1997). Two data sets of near infrared spectra. *Chemometrics and Intelligent Laboratory Systems*, 37(2), 255–259.

Examples

```
data("wheat")

X = wheat$x
y = wheat$protein
n = nrow(wheat$x)

set.seed(1001)
samp.idx = sample(1L:n, round(n * 0.7))
X.cal = X[samp.idx, ]
y.cal = y[samp.idx]
X.test = X[-samp.idx, ]
y.test = y[-samp.idx]
```

Index