# Package ‘smart’

February 20, 2015

**Type**  Package

**Title**  Sparse Multivariate Analysis via Rank Transformation

**Version**  1.0.1

**Date**  2012-08-17

**Author**  Fang Han, Han Liu

**Maintainer**  Fang Han <fhan@jhsph.edu>

**Depends**  R (>= 2.10)

**Imports**  Matrix, gplots, gtools, PMA, elasticnet, pcaPP, igraph

**Suggests**  huge

**Description**  The package `smart` provides a general framework for analyzing (including estimation, feature selection and prediction) and visualize big data. It integrates several novel, efficient and robust data analysis tools, including Transelliptical Component Analysis (TCA), Transelliptical Correlation Estimation (TCE) and Group Nearest Shrunken Centroids (gnsc). We target on high dimensional data analysis (usually d >> n), and exploit computationally efficiently approaches. Results are organized to be visualized properly for users.

**License**  GPL-2

**Repository**  CRAN

**Date/Publication**  2013-10-22 10:50:06

**NeedsCompilation**  no

## R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>smart-package</td>
<td>2</td>
</tr>
<tr>
<td>gnsc.cv</td>
<td>3</td>
</tr>
<tr>
<td>gnsc.train</td>
<td>5</td>
</tr>
<tr>
<td>plot.gnsc</td>
<td>6</td>
</tr>
<tr>
<td>plot.gnsccv</td>
<td>7</td>
</tr>
</tbody>
</table>
Description

A package for Sparse Multivariate Analysis via Rank Transformation

Details

Package: smart
Type: Package
Version: 1.0.0
Date: 2012-08-17
License: GPL-2
LazyLoad: yes

The package "smart" contain three main functions:
(i) "gnsccv" for conducting Group Nearest Shrunken Centroids.
(ii) "TCA" for conducting Transeleiptical Component Analysis.
(iii) "TCE" for conducting Transelliptical Correlation Estimation.

Author(s)

Fang Han, Han Liu
Maintainer: Fang Han<fhan@jhsph.edu>

References

3. Witten, D., Tibshirani, R., and Hastie, T., A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. Biostatistics
A function to cross-validate the Group Nearest Shrunken Centroid Classifier

Description

A function to cross-validate the Group Nearest Shrunken Centroid Classifier produced by gnsc.train

Usage

```r
gnsc.cv(fit, x = NULL, y = NULL, z = NULL, nfold = NULL, folds = NULL, verbose = T)
```

Arguments

- `fit`: The result of a call to gnsc.train
- `x`: The test data matrix (variables in the rows, samples in the columns).
- `y`: The test class labels for samples, must have the same length as the column length of `x`.
- `z`: The test class labels for variables, must have the same length as the row length of `x`.
- `nfold`: Number of cross-validation folds. The default value is the smallest class size.
- `folds`: The fold labels for each sample, must have the same length as `y` and `max(folds) = nfold`. The default value is `sample(1:nfold, n, replace=T)`, here `n` is the sample size.
- `verbose`: If `verbose = FALSE`, tracing information printing is disabled. The default value is `TRUE`.

Details

gnsc.cv carries out a cross-validation for Group Nearest Shrunken Centroid Classifier.
Value

An object with S3 class "gnsccv" is returned:

- `lambda`: A vector of the thresholds tried in the shrinkage
- `nlambda`: The number of thresholds tried in the shrinkage
- `lambda.min`: The index of the threshold which achieves the lowest cross-validation error
- `errors`: The number of cross-validation errors for each threshold value
- `nonzero`: The number of variables that survived the thresholding
- `Thresh.mat`: A list of estimated \( \tilde{\mu}_{mk} \). See Yang, et.al (2012) for details

Author(s)

Fang Han, Han Liu
Maintainer: Fang Han<fhan@jhsph.edu>

References


See Also

`gnsc.train`

Examples

```r
set.seed(120)
x <- matrix(rnorm(1000*20),ncol=20)
y <- sample(c(1:4),size=20,replace=TRUE)
z <- sample(c(1:10),size=1000,replace=TRUE)
fit=gnsc.train(x, col.struc=y, row.struc=z,lambda.max=5, nlambda=20)
fit
plot(fit)
fit.cv=gnsc.cv(fit,x,y,z)
fit.cv
plot(fit.cv)
```
Description
A function to conduct the Group Nearest Shrunken Centroid Classifier

Usage

```r
gnsc.train(x, col.struc = NULL, row.struc = NULL, standardize = T, 
nlambda = NULL, lambda.max = 10, lambda = NULL, verbose = TRUE)
```

Arguments
- `x` The train data matrix (variables in the rows, samples in the columns).
- `col.struc` The train class labels for samples, must have the same length as the column length of `x`.
- `row.struc` The train class labels for variables, must have the same length as the row length of `x`.
- `standardize` Logical value to determine whether to standardize the data. The default value is TRUE.
- `nlambda` The number of thresholding parameters. The default value is 10.
- `lambda.max` The largest lambda value, given the thresholding parameters lambda is not provided by the user.
- `lambda` A sequence of positive numbers to control to determine the thresholding level.
- `verbose` If `verbose = FALSE`, tracing information printing is disabled. The default value is TRUE.

Details
gnsc.train conducts a Group Nearest Shrunken Centroid Classifier.

Value
An object with S3 class "gnsc" is returned:
- `lambda` A vector of the thresholds tried in the shrinkage
- `nlambda` The number of thresholds tried in the shrinkage
- `yhat` A matrix with the estimated sample labels for each thresholding level in each column
- `errors` The number of estimated errors for each threshold value
- `nonzero` The number of variables that survived the thresholding for each thresholding value
- ... System reserved (No specific usage)
plot.gnsc

Author(s)

Fang Han, Han Liu
Maintainer: Fang Han<fhan@jhsph.edu>

References


See Also

gnsc.cv

Examples

```r
set.seed(120)
x <- matrix(rnorm(1000*20),nrow=100)
y <- sample(c(1:4),size=20,replace=TRUE)
z <- sample(c(1:10),size=1000,replace=TRUE)
fit=gnsc.train(x, col.struc=y, row.struc=z,lambda.max=5, nlambda=20)
fit
plot(fit)
```

plot.gnsc

Plot sparsity level information and heatmap from the gnsc.train

Usage

```r
## S3 method for class 'gnsc'
plot(x, which.lambda = NULL, ...)
```

Arguments

- `x`: An object with S3 class "gnsc"
- `which.lambda`: pick one lambda to visualize the intensity heatmap. The default value is median(lambda).
- `...`: System reserved (No specific usage)

Author(s)

Fang Han, Han Liu
Maintainer: Fang Han<fhan@jhsph.edu>
## plot.gnscv

### Plot function for S3 class "gnscv"

#### Description

Plot sparsity level information and heatmap from the gnsc.cv

#### Usage

```r
## S3 method for class 'gnscv'
plot(x, ...)
```

#### Arguments

- **x**: An object with S3 class "gnsccv"
- **...**: System reserved (No specific usage)

#### Author(s)

Fang Han, Han Liu
Maintainer: Fang Han<fhan@jhsph.edu>

#### See Also

- `gnsc.train`
- `plotNgnsccv`

---

## plot.TCA

### Plot function for S3 class "TCA"

#### Description

Plot sparsity level information and projected PCs from the TCA

#### Usage

```r
## S3 method for class 'TCA'
plot(x, pc = NULL, ...)
```
plot.TCE

Arguments

x An object with S3 class "TCA"

pc pick two PCs to visualize the projected principal components. The default value is c(1,2).

... System reserved (No specific usage)

Author(s)

Fang Han, Han Liu
Maintainer: Fang Han<fhan@jhsph.edu>

See Also

TCA

plot.TCE

Plot function for S3 class "TCE"

Description

Plot sparsity level information and 3 typical sparse graphs from the correlation graph path

Usage

## S3 method for class 'TCE'
plot(x, align = FALSE, ...)

Arguments

x An object with S3 class "TCE"

align If align = FALSE, 3 plotted graphs are aligned

... System reserved (No specific usage)

Author(s)

Fang Han, Han Liu
Maintainer: Fang Han<fhan@jhsph.edu>

See Also

TCE
**print.gnsc**  
*Print function for S3 class "gnsc"*

**Description**
Print the information about threshold values, selected variables, fitted errors, predicted values

**Usage**
```r
## S3 method for class 'gnsc'
print(x, ...)
```

**Arguments**
- `x` An object with S3 class "gnsc"
- `...` System reserved (No specific usage)

**Author(s)**
Fang Han, Han Liu  
Maintainer: Fang Han<fhan@jhsph.edu>

**See Also**
- `gnsc.train`

---

**print.gnsccv**  
*Print function for S3 class "gnsccv"*

**Description**
Print the information about threshold values, selected variables, cross-validation errors

**Usage**
```r
## S3 method for class 'gnsccv'
print(x, ...)
```

**Arguments**
- `x` An object with S3 class "gnsccv"
- `...` System reserved (No specific usage)
Author(s)

Fang Han, Han Liu
Maintainer: Fang Han<fhan@jhsph.edu>

See Also

gnsc.cv

print.TCA

Description

Print the information about model, algorithm and results

Usage

## S3 method for class 'TCA'
print(x, ...)

Arguments

x  An object with S3 class "TCA"
... System reserved (No specific usage)

Author(s)

Fang Han, Han Liu
Maintainer: Fang Han<fhan@jhsph.edu>

See Also

TCA
**print.TCE**  
*Print function for S3 class "TCE"*

**Description**
Print the information about the model usage, the graph path length, graph dimension, sparsity level

**Usage**

```r
## S3 method for class 'TCE'
print(x, ...)
```

**Arguments**

- `x`: An object with S3 class "TCE"
- `...`: System reserved (No specific usage)

**Author(s)**
Fang Han, Han Liu  
Maintainer: Fang Han<fhan@jhsph.edu>

**See Also**

- `TCE`

---

**smart-internal**  
*Internal smart functions*

**Description**
Internal smart functions

**Details**
These are not intended for use by users. Please refer to `gnsc.train()`, `gnsc.cv()`, `TCA()`, `TCE()`.

**Author(s)**
Fang Han, Han Liu  
Maintainer: Fang Han<fhan@jhsph.edu>
Transelliptical Component Analysis

Description

A function to conduct Transelliptical Component Analysis

Usage

TCA(x, K, para, method = "kendall", algorithm = "tp", max.iter = 200, verbose = TRUE, eps.conv = 0.001)

Arguments

- **x**: The n by d data matrix or d by d covariance matrix from the input
- **K**: Number of components
- **para**: A vector of length K, indicating the number of sparse loadings.
- **method**: Method to be used to estimating the correlation matrix with 5 options: pearson, ns, npn, spearman and kendall. kendall as default.
- **algorithm**: Algorithm to be used to obtain sparse loadings with 3 options: sp, spca and pmd. tp as default.
- **max.iter**: Maximum number of iterations.
- **verbose**: If verbose = FALSE, tracing information printing is disabled. The default value is TRUE.
- **eps.conv**: Convergence criterion.

Details

PCA and Sparse PCA is sensitive to modeling assumption, outliers, missing values data dependency. We propose an alternative way using rank-based methods including ns, npn, spearman and kendall to approximate the correlation matrix. Details are refered to Han,F. and Liu,H. (2012). Three sparse PCA algorithms are used: truncated power (Yuan, X. and Zhang, T. (2011)), spca(Zou,H., Hastie, T., and Tibshirani, R. (2006)) and pmd (Witten, D., Tibshirani, R., and Hastie, T. (2009)).

Value

- **cov.input**: An indicator of the sample covariance.
- **loadings**: The loadings of the sparse PCs.
- **pev**: An indicator of the sample covariance.
- **PC**: Projected PCs. existing if cov.input=TRUE.
- **method**: The method used in estimating the correlation matrix.
- **algorithm**: The algorithm used in obtaining the sparse loadings.
- **K**: The number of components.
Author(s)
Fang Han, Han Liu
Maintainer: Fang Han<fhan@jhsph.edu>

References
1. Witten, D., Tibshirani, R., and Hastie, T., A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*

Examples
```r
x = matrix(rnorm(20000),100)
fit = tce(x, method = "pearson", nlambda = 20, lambda = NULL)
plot(fit)
```

Description
A function to conduct Transelliptical Correlation Estimation

Usage
```r
TCE(x, method = "pearson", nlambda = NULL, lambda.min.ratio = NULL, lambda = NULL, verbose = TRUE)
```

Arguments
- `x`: The n by d data matrix or d by d covariance matrix from the input
- `method`: Method to be used to estimating the correlation matrix with 5 options: `pearson`, `ns`, `npn`, `spearman` and `kendall`. `kendall` as default.
- `nlambda`: The number of regularization/thresholding paramters. The default value is 20.
- `lambda.min.ratio`: The largest sparsity level for the estimated graphs. The default value is 0.05.
- `lambda`: A sequence of positive numbers for conducting thresholding.
- `verbose`: If `verbose = FALSE`, tracing information printing is disabled. The default value is TRUE.

Details
The correlation graph is estimated by correlation cut-off based on the given thresholding level.
Value

An object with S3 class "TCE" is returned:

- **cov.input**: An indicator of the sample covariance.
- **path**: A list of $k$ by $k$ adjacency matrices of estimated graphs as a graph path corresponding to $\lambda$.
- **sparsity**: The sparsity levels of the graph path.
- **method**: The method used in the correlation graph estimation stage.
- **lambda**: The sequence of thresholding parameters used.

Author(s)

Fang Han, Han Liu
Maintainer: Fang Han<fhan@jhsph.edu>

References


Examples

```r
require(huge)
L = huge.generator(n = 200, d = 80, graph = "hub")
out = tce(L$data, method= "kendall")
out
plot(out)
```
Index

gnsc.cv, 3, 6, 7, 10
gnsc.heatmap (smart-internal), 11
gnsc.icov (smart-internal), 11
gnsc.predict (smart-internal), 11
gnsc.restruc (smart-internal), 11
gnsc.train, 3, 4, 5, 7, 9

minid (smart-internal), 11

plot.gnsc, 6
plot.gnsc.cv, 7
plot.TCA, 7
plot.TCE, 8
print.gnsc, 9
print.gnsc.cv, 9
print.TCA, 10
print.TCE, 11

smart (smart-package), 2
smart-internal, 11
smart-package, 2
smart.npn (smart-internal), 11

TCA, 3, 8, 10, 12
TCE, 3, 8, 11, 13