

Package ‘MGSDA’

March 11, 2021

Type Package

Title Multi-Group Sparse Discriminant Analysis

Version 1.6

Date 2021-03-10

Author Irina Gaynanova

Maintainer Irina Gaynanova <irinag@stat.tamu.edu>

Description Implements Multi-Group Sparse Discriminant Analysis proposal of I.Gaynanova, J.Booth and M.Wells (2016), Simultaneous sparse estimation of canonical vectors in the $p \gg N$ setting, JASA <doi:10.1080/01621459.2015.1034318>.

Imports MASS, stats

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2021-03-11 05:30:02 UTC

R topics documented:

MGSDA-package	1
classifyV	2
cv.dLDA	4
dLDA	5

Index	7
--------------	----------

MGSDA-package	<i>MGSDA: Multi-Group Sparse Discriminant Analysis</i>
---------------	--

Description

Implements Multi-Group Sparse Discriminant Analysis proposal of I.Gaynanova, J.Booth and M.Wells (2016) "Simultaneous sparse estimation of canonical vectors in the $p \gg n$ setting", JASA, 111(514), 696-706.

Details

Package: MGSDA
Type: Package
Version: 1.5
Date: 2021-01-25
License: GPL(>=2)

Three functions for estimation and prediction: `cv.dLDA`, `dLDA` and `classifyV`.

Author(s)

Irina Gaynanova

Maintainer: Irina Gaynanova <irinag@stat.tamu.edu>

References

I.Gaynanova, J.Booth and M.Wells (2016). "Simultaneous sparse estimation of canonical vectors in the $p \gg N$ setting", *JASA*, 111(514), 696-706.

Examples

```
### Example 1
# generate training data
n <- 10
p <- 100
G <- 3
ytrain <- rep(1:G,each=n)
set.seed(1)
xtrain <- matrix(rnorm(p*n*G),n*G,p)
# find matrix of canonical vectors V
V <- dLDA(xtrain,ytrain,lambda=0.1)
sum(rowSums(V)!=0)
# generate test data
m <- 20
set.seed(3)
xtest <- matrix(rnorm(p*m),m,p)
# perform classification
ytest <- classifyV(xtrain,ytrain,xtest,V)
```

classifyV

Classification for MGSDA

Description

Classify observations in the test set using the supplied matrix of canonical vectors V and the training set.

Usage

```
classifyV(Xtrain, Ytrain, Xtest, V, prior = TRUE, tol1 = 1e-10)
```

Arguments

Xtrain	A Nxp data matrix; N observations on the rows and p features on the columns.
Ytrain	A N vector containing the group labels. Should be coded as 1,2,...,G, where G is the number of groups.
Xtest	A Mxp data matrix; M test observations on the rows and p features on the columns.
V	A pxr matrix of canonical vectors that is used to classify observations.
prior	A logical indicating whether to put larger weights to the groups of larger size; the default value is TRUE.
tol1	Tolerance level for the eigenvalues of $V^t W V$. If some eigenvalues are less than tol, the low-rank version of V is used for classification.

Details

For a new observation with the value x , the classification is performed based on the smallest Mahalanobis distance in the projected space:

$$\min_{1 \leq g \leq G} (V^t x - Z_g)(V^t W V)^{-1}(V^t x - Z_g)$$

where Z_g are the group-specific means of the training dataset in the projected space and W is the sample within-group covariance matrix.

If $\text{prior}=\text{T}$, then the above distance is adjusted by $-2 \log \frac{n_g}{N}$, where n_g is the size of group g .

Value

Returns a vector of length M with predicted group labels for the test set.

Author(s)

Irina Gaynanova

References

I.Gaynanova, J.Booth and M.Wells (2016) "Simultaneous Sparse Estimation of Canonical Vectors in the $p \gg N$ setting.", JASA, 111(514), 696-706.

Examples

```
### Example 1
# generate training data
n=10
p=100
G=3
ytrain=rep(1:G, each=n)
```

```

set.seed(1)
xtrain=matrix(rnorm(p*n*G),n*G,p)
# find V
V=dLDA(xtrain,ytrain,lambda=0.1)
sum(rowSums(V)!=0)
# generate test data
m=20
set.seed(3)
xtest=matrix(rnorm(p*m),m,p)
# perform classification
ytest=classifyV(xtrain,ytrain,xtest,V)

```

cv.dLDA

Cross-validation for MGSDA

Description

Chooses optimal tuning parameter lambda for function dLDA based on the m-fold cross-validation mean squared error

Usage

```

cv.dLDA(Xtrain, Ytrain, lambdaval = NULL, nl = 100, msep = 5, eps = 1e-6,
        l_min_ratio = ifelse(n<p,0.1,0.0001),myseed=NULL,prior=TRUE,rho=1)

```

Arguments

Xtrain	A Nxp data matrix; N observations on the rows and p features on the columns
Ytrain	A N vector containing the group labels. Should be coded as 1,2,...,G, where G is the number of groups
lambdaval	Optional user-supplied sequence of tuning parameters; the default value is NULL and cv.dLDA chooses its own sequence
nl	Number of lambda values; the default value is 50
msep	Number of cross-validation folds; the default value is 5
eps	Tolerance level for the convergence of the optimization algorithm; the default value is 1e-6
l_min_ratio	Smallest value for lambda, as a fraction of lambda.max, the data-derived value for which all coefficients are zero; the default value is 0.1 if the number of samples n is larger than the number of variables p, and is 0.001 otherwise.
myseed	Optional specification of random seed for generating the folds; the default value is NULL.
prior	A logical indicating whether to put larger weights to the groups of larger size; the default value is TRUE.
rho	A scalar that ensures the objective function is bounded from below; the default value is 1.

Value

lambdaval	The sequence of tuning parameters used
error_mean	The mean cross-validated number of misclassified observations - a vector of length <code>length(lambdaval)</code>
error_se	The standard error associated with each value of <code>error_mean</code>
lambda_min	The value of tuning parameter that has the minimal mean cross-validation error
f	The mean cross-validated number of non-zero features - a vector of length <code>length(lambdaval)</code>

Author(s)

Irina Gaynanova

References

I.Gaynanova, J.Booth and M.Wells (2016). "Simultaneous sparse estimation of canonical vectors in the $p \gg N$ setting", JASA, 111(514), 696-706.

Examples

```
### Example 1
n=10
p=100
G=3
ytrain=rep(1:G,each=n)
set.seed(1)
xtrain=matrix(rnorm(p*n*G),n*G,p)
# find optimal tuning parameter
out.cv=cv.dLDA(xtrain,ytrain)
# find V
V=dLDA(xtrain,ytrain,lambda=out.cv$lambda_min)
# number of non-zero features
sum(rowSums(V)!=0)
```

dLDA	<i>Estimate the matrix of discriminant vectors using L_1 penalty on the rows</i>
------	---

Description

Solve Multi-Group Sparse Discriminant Analysis problem for the supplied value of the tuning parameter `lambda`.

Usage

```
dLDA(xtrain, ytrain, lambda, Vinit = NULL, eps=1e-6, maxiter=1000, rho=1)
```

Arguments

xtrain	A Nx p data matrix; N observations on the rows and p features on the columns.
ytrain	A N -vector containing the group labels. Should be coded as 1,2,..., G , where G is the number of groups.
lambda	Tuning parameter.
Vinit	A $p \times (G-1)$ optional initial value for the optimization algorithm; the default value is NULL.
eps	Tolerance level for the convergence of the optimization algorithm; the default value is 1e-6.
maxiter	Maximal number of iterations for the optimization algorithm; the default value is 1000.
rho	A scalar that ensures the objective function is bounded from below; the default value is 1.

Details

Solves the following optimization problem:

$$\min_V \frac{1}{2} \text{Tr}(V^t W V + \rho V^t D D^t V) - \text{Tr}(D^t V) + \lambda \sum_{i=1}^p \|v_i\|_2$$

Here W is the within-group sample covariance matrix and D is the matrix of orthogonal contrasts between the group means, both are constructed based on the supplied values of `xtrain` and `ytrain`.

When $G = 2$, the row penalty reduces to vector L_1 penalty.

Value

Returns a $p \times (G-1)$ matrix of canonical vectors V .

Author(s)

Irina Gaynanova

References

I.Gaynanova, J.Booth and M.Wells (2016) "Simultaneous Sparse Estimation of Canonical Vectors in the $p \gg N$ setting", *JASA*, 111(514), 696-706.

Examples

```
# Example 1
n=10
p=100
G=3
ytrain=rep(1:G,each=n)
set.seed(1)
xtrain=matrix(rnorm(p*n*G),n*G,p)
V=dLDA(xtrain,ytrain,lambda=0.1)
sum(rowSums(V)!=0) # number of non-zero rows
```

Index

`classifyV`, [2](#)

`cv.dLDA`, [4](#)

`dLDA`, [5](#)

`MGSDA (MGSDA-package)`, [1](#)

`MGSDA-package`, [1](#)