Package ‘CepLDA’

January 18, 2016

Type Package

Title Discriminant Analysis of Time Series in the Presence of Within-Group Spectral Variability

Version 1.0.0

Date 2015-12-19

URL http://astro.temple.edu/~tuf28410/

Author Zeda Li, Robert Krafty

Maintainer Zeda Li<zeda.li@temple.edu>

Depends R(>= 3.0.1), astsa, MASS, class, mult taper

Description Performs cepstral based discriminant analysis of groups of time series when there exists variability in power spectra from time series within the same group as described in R.T. Krafty (2016) ”Discriminant Analysis of Time Series in the Presence of Within-Group Spectral Variability” Journal of Time Series Analysis.

License GPL (>= 2)

LazyData true

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-01-18 00:35:29

R topics documented:

cep.get ................................................................. 2
cep.lda ................................................................. 3
cep.mtm ................................................................. 5
Lopt.get ................................................................. 6
lperd.mtm ................................................................. 7
plot.ceplda .............................................................. 8
predict.ceplda .......................................................... 8
print.ceplda .......................................................... 10
r.cond.ar2 .............................................................. 10
recon ................................................................. 11
cep.get

Index

Description
Returns a data frame containing raw cepstra coefficients and the group membership from multiple time series.

Usage
cep.get(y, x, nw, k)

Arguments
- y: n-vector indicating group membership
- x: N by n matrix containing n time series, each with length N.
- nw: Width of tapers used in multitaper spectral estimation. Default is set to 4
- k: Number of tapers used in multitaper spectral estimation. Default is set to 7

Value
dhat: Data frame containing group information and raw cepstral coefficients.

Author(s)
Zeda Li <<zeda.li@temple.edu>>

References

See Also
predict.ceplda, lopt.get

Examples
```r
## Simulate dataset
nj = 50  #number of series in training data
N = 500  #length of time series
data1 <- r.cond.ar2(N=N, nj=nj, r.phi1=c(.01,.7), r.phi2=c(-.12,-.06), r.sig2=c(.3,3))
data2 <- r.cond.ar2(N=N, nj=nj, r.phi1=c(.5,1.2), r.phi2=c(-.36,-.25), r.sig2=c(.3,3))
data3 <- r.cond.ar2(N=N, nj=nj, r.phi1=c(.9,1.5), r.phi2=c(-.56,-.75), r.sig2=c(.3,3))
dat <- cbind(data1$x, data2$x, data3$x)
y <- c(rep(1,nj), rep(2,nj), rep(3,nj))
data.cep <- cep.get(y, dat, 4, 7)
dim(data.cep)
```
### Description
The main program.

### Usage

```r
cep.llda(y, x, xNew, L, mcep, nw, k, cv, tol)
```

### Arguments

- **y**: n-vector indicating group membership of training time series.
- **x**: \(N \times n\) matrix containing \(n\) training time series each with length \(N\).
- **xNew**: \(N \times n_{New}\) matrix, containing \(n_{New}\) time series whose memberships are predicted.
- **L**: Number of cepstral coefficients used in the lda. If FALSE, cross-validation is used for the data driven selection of \(L\). Default is FALSE.
- **mcep**: Maximum number of cepstral coefficients considered. Default is set to 10.
- **nw**: Width of tapers used in multitaper spectral estimation. Default is set to 4.
- **k**: Number of tapers used in multitaper spectral estimation. Default is set to 7.
- **cv**: If TRUE, returns results (classes and posterior probabilities) for leave-one-out cross-validation. Note that, if the prior is estimated, the proportions in the whole dataset are used. As with the standard lda function, if used, prediction on a test data set cannot be done and weight functions are not produced (similar to the predict.lda). Default is FALSE.
- **tol**: Tolerance to decide if a matrix is singular; it will reject variables and linear combinations of unit-variance variables whose variance is less than \(tol^2\).

### Value

List with 5 elements

- **c.lda**: lda output on the cepstral scale. Similar to output of lda(MASS) function.
- **cep.data**: Data frame containing cepstral coefficients and group information from training data.
- **lopt**: Number of cepstral coefficients used.
- **lspec**: Estimated log-spectral weight functions.
- **predict**: Results of classification. If external data xNew is supplied, these data are classified. If not, biased classification of the training data x is returned. For unbiased leave-out-one cross-validated classification of training data, use cv=TRUE.
Author(s)

Zeda Li <<zeda.li@temple.edu>>; Robert Krafty <<rkrafty@pitt.edu>>

References


See Also

`predict.ceplda`, `plot.ceplda`, `print.ceplda`, `Lopt.get`

Examples

```r
## Simulate training data
nj = 50  # number of series in training data
N = 500  # length of time series
traindata1 <- r.cond.ar2(N=N, nj=nj, r.phi1=c(.01,.7), r.phi2=c(-.12,-.06), r.sig2=c(.3,3))
traindata2 <- r.cond.ar2(N=N, nj=nj, r.phi1=c(.5,1.2), r.phi2=c(-.36,-.25), r.sig2=c(.3,3))
traindata3 <- r.cond.ar2(N=N, nj=nj, r.phi1=c(.9,1.5), r.phi2=c(-.56,-.75), r.sig2=c(.3,3))
train <- cbind(traindata1[,], traindata2[,], traindata3[,])
y <- c(rep(1,nj),rep(2,nj),rep(3,nj))

## Fit the discriminant analysis
fit <- ceplda(y, train)
fit  # displays group means and cepstral weight functions

## Discriminant plot
plot(fit)

## Plot log-spectral weights
par(mfrow=c(1,2))
plot(fit$lspec$frq, fit$lspec$dsc[,1], type='l', xlab="frequency", ylab="log-spectral weights")
plot(fit$lspec$frq, fit$lspec$dsc[,2], type='l', xlab="frequency", ylab="log-spectral weights")

## Bias classification of training data
mean(fit$predict$class == y)  # classification rate
table(y, fit$predict$class)

## Fit the discriminant analysis while classifying training data via cross-validation
fit.cv <- cep.lda(y, train, cv=TRUE)
mean(fit.cv$predict$class == y)  # classification rate
table(y, fit.cv$predict$class)

## Simulate test data
testdata1 <- r.cond.ar2(N=N, nj=nj, r.phi1=c(.01,.7), r.phi2=c(-.12,-.06), r.sig2=c(.3,3))
testdata2 <- r.cond.ar2(N=N, nj=nj, r.phi1=c(.5,1.2), r.phi2=c(-.36,-.25), r.sig2=c(.3,3))
testdata3 <- r.cond.ar2(N=N, nj=nj, r.phi1=c(.9,1.5), r.phi2=c(-.56,-.75), r.sig2=c(.3,3))
test <- cbind(testdata1[,], testdata2[,], testdata3[,])
yTest <- c(rep(1,nj),rep(2,nj),rep(3,nj))

## Fit discriminant analysis and classify new data
```
cep.mtm

Multitaper Estimation of Cepstral Coefficients and the Log-Spectrum

Description

Returns multitaper estimated cepstra coefficients and log-spectrum for univariate time series.

Usage

cep.mtm(x, nw, k)

Arguments

x
Univariate time series of length N.

nw
Width of tapers. Default is set to 4

k
Number of tapers. Default is set to 7

Value

a list with 4 elements

quef
Quefencies.

cep
Raw cepstra coefficients from 0:(N-1)

freq
Frequencies between 0 and 1

lspec
Log-spectrum.

Author(s)

Robert Krafty <<rkrafty@pitt.edu>>

Examples

## simulate a time series
N = 500 # length of each series
dat <- r.cond.ar2(N=N, nj=1, r.phi1=c(.01,.7), r.phi2=c(-.12,-.06), r.sig2=c(.3,.3))$X

## Fit multiaper
cp <- cep.mtm(dat)

## Plot the cepstral coefficients
plot(cp$quef, cp$cep)

## Plot the log spectrum
plot(cp$freq, cp$lspec, type="l")
**Lopt.get**  
*Optimal Choice of L*

**Description**

Data driven selection of the number of cepstral coefficients \((L)\) via leave-one-out cross-validation.

**Usage**

```r
Lopt.get(data,mcep)
```

**Arguments**

- **data**: Data frame containing cepstral coefficients and group information. Obtained from `cep.get` or `cep.lda`.
- **mcep**: Maximum number of cepstral coefficient considered. Default is set to 10

**Value**

- **Lopt**: Optimal number of cepstral coefficients

**Author(s)**

Robert Krafty <rkrafty@pitt.edu>

**References**


**See Also**

`cep.lda`, `cep.get`

**Examples**

```r
## Simulate data
ntrain = 50  # number of series in training data
Nlength = 500  # length of each series
set.seed(2016)
traindata1 <- r.cond.ar2(N=Nlength,nj=ntrain,r.phi1=c(.01,.7),r.phi2=c(-.12,-.06),r.sig2=c(.3,3))
traindata2 <- r.cond.ar2(N=Nlength,nj=ntrain,r.phi1=c(.5,1.2),r.phi2=c(-.36,-.25),r.sig2=c(.3,3))
traindata3 <- r.cond.ar2(N=Nlength,nj=ntrain,r.phi1=c(.9,1.5),r.phi2=c(-.56,-.75),r.sig2=c(.3,3))
train <- cbind(traindata1[,X],traindata2[,X],traindata3[,X])
## group information
y <- c(rep(1,ntrain),rep(2,ntrain),rep(3,ntrain))
dat <- cep.get(y,train)
Lopt.get(dat,10)
```
**Description**

Obtain multivariate log-periodogram. Not often used by itself, but is used as part of cep.mtm.

**Usage**

lperd.mtm(x, nw, k)

**Arguments**

- **x**  
  N by n matrix containing n training time series each with length N.
- **nw**  
  Width of tapers used in multitaper spectral estimation.
- **k**  
  Number of tapers used in multitaper spectral estimation.

**Value**

- **freq**  
  Fourier frequencies from 0 to 1.
- **lspec**  
  Log-periodogram.
- **spec**  
  Periodogram on the natural scale.

**Author(s)**

Robert Krafty <rkrafty@pitt.edu>

**References**


**See Also**

cep.mtm, cep.lda
**plot.ceplda**

*Plot Method for Class 'ceplda'*

**Description**

Plots a set of data on one, two or more linear discriminants.

**Usage**

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

**Arguments**

- `x`: an object of class "ceplda".
- `...`: additional arguments.

**Details**

This function is a method for the generic function `plot()` for class "ceplda". It can be invoked by calling `plot(x)` for an object `x` of the appropriate class, or directly by calling `plot.ceplda(x)` regardless of the class of the object. Details usage of this function is equivalent to `plot.lda(MASS)`.

**See Also**

- `cep.lda`, `plot.lda`

---

**predict.ceplda**

*Classify Multivariate Time Series*

**Description**

Classify time series. Run as part of `cep.lda`, and can be run seperately after running `cep.lda`.

**Usage**

```r
## S3 method for class 'ceplda'
predict(object, newdata, ...)
```

**Arguments**

- `object`: Object of class "ceplda".
- `newdata`: Data frame of cases to be classified. Data frame can be obtained by `cep.get`.
- `...`: argument based from or to other methods.
predict.ceplda

Details

This function is a method for the generic function predict() for class "ceplda". It can be invoked by calling predict(x) for an object x of the appropriate class, or directly by calling predict.lda(x) regardless of the class of the object. Details usage of this function is similar to predict.lda(MASS).

Value

List with components

- **class**: Classification result (a factor).
- **posterior**: Posterior class probabilities.
- **x**: Scores from test data.

See Also

cpl.lda, predict.lda

Examples

```r
## See cep.lda for predicting new data simultaneously while fitting a model to training data.
## Below is predicting new data after fitting a model to the training data.

## Simulate training data
def n = 50  # number of series in training data
def N = 500  # length of time series
traindata1 = r.cond.ar2(N=n, n=n, r.phi1=c(0.01, 0.7), r.phi2=c(-1.2, -0.06), r.sig2=c(0.3, 3))
traindata2 = r.cond.ar2(N=n, n=n, r.phi1=c(0.5, 1.2), r.phi2=c(-0.36, -0.25), r.sig2=c(0.3, 3))
traindata3 = r.cond.ar2(N=n, n=n, r.phi1=c(0.9, 1.5), r.phi2=c(-0.56, -0.75), r.sig2=c(0.3, 3))
train <- cbind(traindata1$x, traindata2$x, traindata3$x)
y <- c(rep(1, n), rep(2, n), rep(3, n))

## Fit the discriminant analysis
fit <- cep.lda(y, train)

## Simulate test data
testdata1 = r.cond.ar2(N=N, n=n, r.phi1=c(0.01, 0.7), r.phi2=c(-1.2, -0.06), r.sig2=c(0.3, 3))
testdata2 = r.cond.ar2(N=N, n=n, r.phi1=c(0.5, 1.2), r.phi2=c(-0.36, -0.25), r.sig2=c(0.3, 3))
testdata3 = r.cond.ar2(N=N, n=n, r.phi1=c(0.9, 1.5), r.phi2=c(-0.56, -0.75), r.sig2=c(0.3, 3))
test <- cbind(testdata1$x, testdata2$x, testdata3$x)
yTest <- c(rep(1, n), rep(2, n), rep(3, n))

## Classify new data
pre <- predict(fit, cpl.get(yTest, test))
mean(pre$class == yTest)
table(yTest, pre$class)
```
**print.ceplda**  
*Print Method for Class 'ceplda'*

**Description**

Print the results from cep.lda

**Usage**

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

**Arguments**

- `x`: an object of class "ceplda".
- `...`: additional arguments.

**See Also**

cep.lda

---

**r.cond.ar2**  
*Generate Random AR(2) Time Series*

**Description**

Simulates multiple AR(2) time series.

**Usage**

```r
r.cond.ar2(N,nj,r.phi1,r.phi2,r.sig2)
```

**Arguments**

- `N`: Length of the series
- `nj`: Number of series generated.
- `r.phi1`: Range of first AR order coefficient. It is a vector contains minimum and maximum possible coefficients.
- `r.phi2`: Range of second AR order coefficient. It is a vector contains minimum and maximum possible coefficients.
- `r.sig2`: Range of conditional innovation variances. It is a vector contains minimum and maximum possible variances.
recon

Value

- a list with 2 elements
  - X \( N \times nj \) matrix of time series
  - cep \( 3 \times nj \) matrix of parameters (\( \phi_1, \phi_2, \sigma_2 \))

Author(s)

Robert Krafty <<rkrafty@pitt.edu>>

References


See Also

cep.lda

Examples

```r
## Simulate data
nj = 50  #number of series in training data
N = 500  #length of time series
data1 <- r.cond.ar2(N=N,nj=nj,r.phi1=c(.01,.7),r.phi2=c(-.12,-.06),r.sig2=c(.3,3))
data2 <- r.cond.ar2(N=N,nj=nj,r.phi1=c(.5,1.2),r.phi2=c(-.36,-.25),r.sig2=c(.3,3))
data3 <- r.cond.ar2(N=N,nj=nj,r.phi1=c(.9,1.5),r.phi2=c(-.56,-.75),r.sig2=c(.3,3))
data <- cbind(data1$X,data2$X,data3$X)
y <- c(rep(1,nj),rep(2,nj),rep(3,nj))
```

---

recon

*Convert Cepstral Coefficients into Log-Spectra.*

Description

Returns a log-spectrum at a given set of frequencies from a cepstrum.

Usage

```r
recon(wts,fqs)
```

Arguments

- wts  
  Cepstral coefficients.
- fqs  
  Frequencies to evaluate the log-spectrum.

Value

- dsc  
  The log-spectrum.
Author(s)

Robert Krafty <<rkrafty@pitt.edu>>

Examples

```r
## Simulate dataset
nj = 50  # number of series in training data
N = 500  # length of time series
data1 <- r.cond.ar2(N=N, nj=nj, r.phi1=c(.01,.7), r.phi2=c(-.12, -.06), r.sig2=c(.3,3))
data2 <- r.cond.ar2(N=N, nj=nj, r.phi1=c(.5, 1.2), r.phi2=c(-.36, -.25), r.sig2=c(.3,3))
data3 <- r.cond.ar2(N=N, nj=nj, r.phi1=c(.9, 1.5), r.phi2=c(-.56, -.75), r.sig2=c(.3,3))
dat <- cbind(data1, data2, data3)
y <- c(rep(1, nj), rep(2, nj), rep(3, nj))
data.cep <- cep.get(y, dat, 4, 7)

## Convert cepstral coefficients into log-spectra
frqs <- seq(from=0, to=.5, by=1/(dim(data.cep)[2]-1))
lspec <- matrix(0, dim(data.cep)[1], length(frqs))
## rows of lspec matrix contains log-spectra
for(i in 1:dim(data.cep)[1]){  
  lspec[i,] <- recon(data.cep[,i], frqs)
}
```
Index

cep.get, 2, 6
cep.lda, 3, 6–11
cep.mtm, 5, 7
Lopt.get, 2, 4, 6
lperd.mtm, 7
plot.ceplda, 4, 8
plot.lda, 8
predict.ceplda, 2, 4, 8
predict.lda, 9
print.ceplda, 4, 10
r.cond.ar2, 10
recon, 11