Package 'SpatMCA'

April 2, 2018

Type Package

Title Regularized Spatial Maximum Covariance Analysis

Version 1.0.1.0

Date 2018-04-01

URL https://github.com/egpivo/SpatMCA

BugReports https://github.com/egpivo/SpatMCA/issues

Description Provide regularized maximum covariance analysis incorporating smoothness, sparseness and orthogonality of couple patterns by using the alternating direction method of multipliers algorithm. The method can be applied to either regularly or irregularly spaced data, including 1D, 2D, and 3D (Wang and Huang, 2017 <doi:10.1002/env.2481>).

License GPL-2

Imports Rcpp, RcppParallel (>= 0.11.2), fields, MASS

LinkingTo Rcpp, RcppArmadillo, RcppParallel

SystemRequirements GNU make

NeedsCompilation yes

Author Wen-Ting Wang [aut, cre], Hsin-Cheng Huang [aut]

Maintainer Wen-Ting Wang <egpivo@gmail.com>

Repository CRAN

Date/Publication 2018-04-02 09:41:07 UTC

R topics documented:

SpatMCA-package 2
spatmca 2

Index **7**

`SpatMCA-package` *Regularized Spatial Maximum Covariance Analysis*

Description

A new regularization approach to estimate the leading coupled patterns via smoothness and sparseness penalties for spatial bivariate data that may be irregularly located in space.

Details

Package:	SpatMCA
Type:	Package
Version:	1.0.1.0
Date:	2018-04-01
License:	GPL version 2 or newer

Author(s)

Wen-Ting Wang <egpivo@gmail.com> and Hsin-Cheng Huang <hchuang@stat.sinica.edu.tw>

`spatmca` *Regularized spatial MCA*

Description

Produce spatial coupled patterns at the designated locations according to the specified tuning parameters or the tuning parameters selected by M-fold cross-validation.

Usage

```
spatmca(x1, x2, Y1, Y2, M = 5, K = NULL, K.select = ifelse(is.null(K),TRUE,FALSE),
tau1u = NULL, tau2u = NULL, tau1v = NULL, tau2v = NULL, x1_new = NULL, x2_new = NULL,
center = TRUE, plot.cv = FALSE, maxit = 100, thr = 1e-04, fullselect  = FALSE)
```

Arguments

`x1` Location matrix ($p \times d$) correponding to Y1. Each row is a location. $d = 1, 2$ is the dimension of locations.

`x2` Location matrix ($q \times d$) correponding to Y2. Each row is a location.

`Y1` Data matrix ($n \times p$) of the first variable stores the values at p locations with sample size n.

`Y2`	Data matrix ($n \times q$) of the second variable stores the values at q locations with sample size n.
`M`	Optional number of folds; default is 5.
`K`	Optional user-supplied number of coupled patterns; default is NULL. If K is NULL or K.select is TRUE, K is selected automatically.
`K.select`	If TRUE, K is selected automatically; otherwise, K.select is set to be user-supplied K. Default depends on user-supplied K.
`tau1u`	Optional user-supplied numeric vector of a nonnegative smoothness parameter sequence correponding to Y1. If NULL, 10 tau1u values in a range are used.
`tau2u`	Optional user-supplied numeric vector of a nonnegative sparseness parameter sequence correponding to Y1. If NULL, 10 tau2u values in a range are use.
`tau1v`	Optional user-supplied numeric vector of a nonnegative smoothness parameter sequence correponding to Y2. If NULL, 10 tau1v values in a range are used.
`tau2v`	Optional user-supplied numeric vector of a nonnegative sparseness parameter sequence correponding to Y2. If NULL, 10 tau2v values in a range are use.
`x1_new`	New location matrix correponding to Y1. If NULL, it is x1.
`x2_new`	New location matrix correponding to Y2. If NULL, it is x2.
`center`	If TRUE, center the columns of Y1 and Y2. Default is TRUE.
`plot.cv`	If TRUE, plot the cv values. Default is FALSE.
`maxit`	Maximum number of iterations. Default value is 100.
`thr`	Threshold for convergence. Default value is 10^{-4}.
`fullselect`	If TRUE, The K-fold CV performs to select 4 tuning parameters simultaneously. Default value is FALSE.

Details

The optimization problem is

$$\max_{\boldsymbol{U},\boldsymbol{V}} \frac{1}{n}\mathrm{tr}(\boldsymbol{U}'\boldsymbol{Y}_1'\boldsymbol{Y}_2\boldsymbol{V}) - \tau_{1u}\mathrm{tr}(\boldsymbol{U}'\boldsymbol{\Omega}_1\boldsymbol{U}) - \tau_{2u}\sum_{k=1}^{K}\sum_{j=1}^{p}|u_{jk}| - \tau_{1v}\mathrm{tr}(\boldsymbol{V}'\boldsymbol{\Omega}_2\boldsymbol{V}) - \tau_{2v}\sum_{k=1}^{K}\sum_{j=1}^{q}|v_{jk}|,$$

subject to $\boldsymbol{U}'\boldsymbol{U} = \boldsymbol{V}'\boldsymbol{V} = \boldsymbol{I}_K$, where $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$ are two data matrices, $\boldsymbol{\Omega}_1$ and $\boldsymbol{\Omega}_2$ are two smoothness matrix, $\boldsymbol{V} = \{v_{jk}\}$, and $\boldsymbol{U} = \{u_{jk}\}$.

Value

`Uestfn`	Estimated patterns for Y1 at the new locations, x1_new.
`Vestfn`	Estimated patterns for Y2 at the new locations, x2_new.
`Dest`	Estimated singular values.
`crosscov`	Estimated cross-covariance matrix between Y1 and Y2.
`stau1u`	Selected tau1u.
`stau2u`	Selected tau2u.
`stau1v`	Selected tau1v.

stau2v	Selected tau2v.
cv1	cv socres for tau1u and tau1v when fullselect is FALSE.
cv2	cv socres for tau2u and tau2v when fullselect is FALSE.
cvall	cv socres for tau1u, tau2u, tau1v and tau2v when fullselect is TRUE.
tau1u	Sequence of tau1u-values used in the process.
tau2u	Sequence of tau2u-values used in the process.
tau1v	Sequence of tau1v-values used in the process.
tau2v	Sequence of tau2v-values used in the process.

Author(s)

Wen-Ting Wang and Hsin-Cheng Huang

References

Wang, W.-T. and Huang, H.-C. (2017). Regularized spatial maximum covariance analysis, to submit.

Examples

```
############### 1D: regular locations ################################
p <- q <- 20
n <- 100
x1 <- matrix(seq(-7, 7, length = p), nrow = p, ncol = 1)
x2 <- matrix(seq(-7, 7, length = q), nrow = q, ncol = 1)
u <- exp(-x1^2)/norm(exp(-x1^2), "F")
v <- exp(-(x2 - 2)^2)/norm(exp(-(x2 - 2)^2), "F")
Sigma <- array(0, c(p + q, p + q))
Sigma[1:p, 1:p] <- diag(p)
Sigma[(p + 1):(p + q), (p + 1):(p + q)] <- diag(p)
Sigma[1:p, (p + 1):(p + q)] <- u%*%t(v)
Sigma[(p + 1):(p + q), 1:p] <- t(Sigma[1:p, (p + 1):(p + q)])
noise <- MASS::mvrnorm(n, mu = rep(0, p + q), Sigma = 0.001*diag(p + q))
Y <- MASS:::mvrnorm(n, mu = rep(0, p + q), Sigma = Sigma) + noise
Y1 <- Y[,1:p]
Y2 <- Y[,-(1:p)]
cv1 <- spatmca(x1, x2, Y1, Y2)
par(mfrow=c(2,1))
plot(x1, cv1$Uestfn[,1], type='l', main="1st pattern for Y1")
plot(x1, cv1$Vestfn[,1], type='l', main="1st pattern for Y2")
## Not run:
### 1D: artificial irregular locations
rm_loc1 <- sample(1:p, 10)
rm_loc2 <- sample(1:q, 10)
x1_rm <- x1[-rm_loc1]
x2_rm <- x2[-rm_loc2]
Y1_rm <- Y1[,-rm_loc1]
Y2_rm <- Y2[,-rm_loc2]
x1_new <- as.matrix(seq(-7, 7, length = 100))
x2_new <- as.matrix(seq(-7, 7, length = 50))
```

```
cv2 <- spatmca(x1 = x1_rm, x2 = x2_rm, Y1 = Y1_rm, Y2 = Y2_rm, x1_new = x1_new, x2_new = x2_new)
par(mfrow=c(2,1))
plot(x1_new, cv2$Uestfn[,1], type='l', main="1st pattern for Y1")
plot(x2_new, cv2$Vestfn[,1], type='l', main="1st pattern for Y2")

###############################2D real data#########################################
# Daily 8-hour ozone averages and maximum temperature obtained from 28 monitoring
# sites of NewYork, USA. It is of interest to see the relationship between the ozone
# and the temperature through the coupled patterns.
####################################################################################
library(spTimer)
library(pracma)
data(NYdata)
NYsite <- unique(cbind(NYdata[,1:3]))
date <- as.POSIXct(seq(as.Date("2006-07-01"), as.Date("2006-08-31"), by = 1))
cMAXTMP<- matrix(NYdata[,8], 62, 28)
oz <- matrix(NYdata[,7], 62, 28)
rmna <- !colSums(is.na(oz))
temp <- detrend(matrix(cMAXTMP[,rmna], nrow = nrow(cMAXTMP)), "linear")
ozone <- detrend(matrix(oz[,rmna], nrow = nrow(oz)), "linear")
x1 <- NYsite[rmna, 2:3]
cv <- spatmca(x1, x1, temp, ozone)
par(mfrow=c(2,1))
quilt.plot(x1, cv$Uestfn[,1], xlab = "longitude", ylab = "latitude",
main = "1st spatial pattern for temperature")
 map(database = "state", regions = "new york", add = T)
quilt.plot(x1, cv$Vestfn[,1], xlab = "longitude", ylab = "latitude",
 main = "1st spatial pattern for ozone")
map(database = "state", regions = "new york", add = T)

###Time series for the coupled patterns
tstemp <- temp%*%cv$Uestfn[,1]
tsozone <- ozone%*%cv$Vestfn[,1]
corr <- cor(tstemp, tsozone)
plot(date, tstemp/sd(tstemp), type='l', main = "Time series", ylab = "", xlab = "month")
lines(date, tsozone/sd(tsozone),col=2)
legend("bottomleft", c("Temperature (standardized)", "Ozone (standardized)"), col = 1:2, lty = 1:1)
mtext(paste("Pearson's correlation = ", round(corr,3)), 3)

###new locations
new_p = 50
x_lon <- seq(-80, -72, length = new_p)
x_lat <- seq(41, 45, length = new_p)
xx_new <- as.matrix(expand.grid(x = x_lon, y = x_lat))
cv_new <- spatmca(x1,x1, temp, ozone, K = cv$Khat, tau1u = cv$stau1u, tau1v =
cv$stau1v, tau2u = cv$stau2u, tau2v = cv$stau2v, x1_new = xx_new, x2_new = xx_new)
par(mfrow=c(2,1))
quilt.plot(xx_new, cv_new$Uestfn[,1], nx = new_p, ny = new_p, xlab = "longitude",
ylab = "latitude", main = "1st spatial pattern for temperature")
map(database="county", regions="new york", add = T)
map.text("state", regions="new york", cex = 2, add = T)
quilt.plot(xx_new, cv_new$Vestfn[,1], nx = new_p, ny = new_p,  xlab = "longitude",
ylab = "latitude", main = "2nd spatial pattern for ozone")
```

```
map(database="county", regions="new york", add = T)
map.text("state", regions="new york", cex = 2, add = T)

###### 3D: regular locations
n <- 200
x <- y <- z <- as.matrix(seq(-7, 7, length = 8))
d <- expand.grid(x, y, z)
u_3D <- v_3D <- exp(-d[,1]^2 - d[,2]^2 -d[,3]^2)
p <- q <- 8^3
Sigma_3D <- array(0, c(p + q, p + q))
Sigma_3D[1:p, 1:p] <- diag(p)
Sigma_3D[(p + 1):(p + q), (p + 1):(p + q)] <- diag(p)
Sigma_3D[1:p, (p + 1):(p + q)] <- u_3D%*%t(v_3D)
Sigma_3D[(p + 1):(p + q), 1:p] <- t(Sigma_3D[1:p, (p + 1):(p + q)])

noise_3D <- MASS::mvrnorm(n, mu = rep(0, p + q), Sigma = 0.001*diag(p + q))
Y_3D <- MASS:::mvrnorm(n, mu = rep(0, p + q), Sigma = Sigma_3D) + noise_3D
Y1_3D <- Y_3D[,1:p]
Y2_3D <- Y_3D[,-(1:p)]
cv_3D <- spatmca(d, d, Y1_3D, Y2_3D)

library(plot3D)
library(RColorBrewer)
cols <- colorRampPalette(brewer.pal(9,'Blues'))(10)
isosurf3D(x, y, z, colvar = array(cv_3D$Uestfn[,1], c(8, 8, 8)),
            level= seq(min(cv_3D$Uestfn[,1]), max(cv_3D$Uestfn[,1]), length=10),
            ticktype = "detailed",
            colkey = list(side = 1),
            col = cols,
            main = "1st estimated pattern for Y1")

isosurf3D(x, y, z, colvar = array(cv_3D$Vestfn[,1], c(8, 8, 8)),
            level= seq(min(cv_3D$Vestfn[,1]), max(cv_3D$Vestfn[,1]), length=10),
            ticktype = "detailed",
            colkey = list(side = 1),
            col = cols,
            main = "1st estimated pattern for Y2")
## End(Not run)
```

Index

spatmca, 2
SpatMCA-package, 2