Package ‘probFDA’

May 11, 2015

Type Package

Title Probabilistic Fisher Discriminant Analysis

Version 1.0.1

Date 2015-05-11

Author Charles Bouveyron & Camille Brunet

Maintainer Charles Bouveyron `<charles.bouveyron@parisdescartes.fr>`

Depends MASS

Description Probabilistic Fisher discriminant analysis (pFDA) is a probabilistic version of the popular and powerful Fisher linear discriminant analysis for dimensionality reduction and classification.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2015-05-11 23:56:13

R topics documented:

`probFDA-package` *A Probabilistic Version of Fisher Linear Discriminant Analysis*

Description

Probabilistic Fisher discriminant analysis (pFDA) is a probabilistic version of the popular and powerful Fisher linear discriminant analysis for dimensionality reduction and classification. PFDA overcomes the known limitations of FDA in the contexts of label noise and sparse labeled data. To this end, pFDA relaxes the homoscedastic assumption on the class covariance matrices and adds a term to explicitly model the non-discriminative information. The pFDA method works at least as well as the traditional FDA method (even better in most cases) in standard situations and it clearly improves the modeling and the prediction when the dataset is subject to label noise and/or sparse labels. The practitioner may therefore replace without prejudice FDA by pFDA for its daily use.

Details

Package:	pFDA
Type:	Package
Version:	1.0
Date:	2015-01-26
License:	GPL-v2

Author(s)

Charles Bouveyron and Camille Brunet

Maintainer: Charles Bouveyron <charles.bouveyron@parisdescartes.fr>

References

C. Bouveyron and C. Brunet, Probabilistic Fisher discriminant analysis: A robust and flexible alternative to Fisher discriminant analysis, Neurocomputing, vol. 90 (1), pp. 12-22, 2012.

See Also

`lda`

Examples

```
palette(c("#E41A1C","#377EB8","#4DAF4A"))

# Simulation of data
n = 900; p = 25
n1 = 1/3*n; n2 = 1/3*n; n3 = 1/3*n;
S1 = diag(2)
S2 = rbind(c(1,-0.95),c(-0.95,1))
S3 = rbind(c(2,0),c(0,0.05))
m1 = c(0,0); m2 = c(0,2); m3 = c(2,0)
X = rbind(mvrnorm(n1,m1,S1),mvrnorm(n2,m2,S2),mvrnorm(n3,m3,S3))
Q = qr.Q(qr(mvrnorm(p,mu=rep(0,p),Sigma=diag(25,p))))
B = mvrnorm(nrow(X),rep(0,p-2),0.1*diag(rep(p-2,p-2)))
```

```
X = crossprod(t(cbind(X,B)),Q)
cls = rep(c(1,2,3),c(n1,n2,n3))

# Cross-validation
nbrep = 10
CCR = matrix(NA,2,nbrep)
for (i in 1:nbrep){
  ind = sample(n)[1:(3/5*n)]
  lda.c = lda(X[ind,],cls[ind])
  res = predict(lda.c,X[-ind,])
  CCR[1,i] = sum(res$cl==cls[-ind])/length(cls[-ind])
  prms = pfda(X[ind,],cls[ind],model=c('DkBk','DB','AkB','AB'),crit='bic',display=TRUE)
  res = predict(prms,X[-ind,])
  CCR[2,i] = sum(res$cl==cls[-ind])/length(cls[-ind])
}

# Display results
split.screen(c(2,1))
split.screen(c(1,3), screen = 1)
screen(3)
plot(predict(princomp(X)),col=cls,pch=(17:19)[cls],main='PCA')
screen(4)
plot(crossprod(t(X),lda(X,cls)$scaling),col=cls,pch=(17:19)[cls],main='LDA')
screen(5)
plot(crossprod(t(X),pfda(X,cls,model='DkBk')$V),col=cls,pch=(17:19)[cls],main='PFDA',
  xlab='LD1',ylab='LD2')
screen(2)
boxplot(t(CCR),names=c('LDA','PFDA'),col=c(1,2),ylab="CCR",
  main='CV correct classification rate')
```

pfda *A Probabilistic Version of Fisher Linear Discriminant Analysis*

Description

Probabilistic Fisher discriminant analysis (pFDA) is a probabilistic version of the popular and powerful Fisher linear discriminant analysis for dimensionality reduction and classification. PFDA overcomes the known limitations of FDA in the contexts of label noise and sparse labeled data. To this end, pFDA relaxes the homoscedastic assumption on the class covariance matrices and adds a term to explicitly model the non-discriminative information. The pFDA method works at least as well as the traditional FDA method (even better in most cases) in standard situations and it clearly improves the modeling and the prediction when the dataset is subject to label noise and/or sparse labels. The practitioner may therefore replace without prejudice FDA by pFDA for its daily use.

Usage

```
pfda(Y, cls, model = "AkjBk", crit = "bic", cv.fold = 10, kernel = "", display = FALSE)
```

Arguments

`Y` The data.

`cls` The labels.

`model` The model name. Possible model names are `'DkBk'`, `'DkB'`, `'DBk'`, `'DB'`, `'AkjBk'`, `'AkjB'`, `'AkBk'`, `'AkB'`, `'AjBk'`, `'AjB'`, `'ABk'`, `'AB'` or `'all'`. The `'all'` option allows to try all models. The default is `'AkjBk'`.

`crit` The model selection criteria. Possible criteria are BIC (`'bic'`) or cross-validation (`'cv'`). The default is `'bic'`.

`cv.fold` In case of cross-validation, the number of cross-validation folds.

`kernel` The kernel option allows to do the classification in a feature space generated by a kernel. This could be especially useful when n < p. The possible options are `''`, `'linear'`, `'sigmoid'` or `'rbf'`. The default is no kernel (`''`).

`display` A display option. The default is `FALSE`.

Value

A list of class `pfda` is returned:

`prms` All estimated model parameters.

`V` The estimated loading matrix.

`bic` The BIC value.

`ll` The log-likelihood value.

`K` The estimated loading matrix.

Author(s)

Charles Bouveyron and Camille Brunet

References

C. Bouveyron and C. Brunet, Probabilistic Fisher discriminant analysis: A robust and flexible alternative to Fisher discriminant analysis, Neurocomputing, vol. 90 (1), pp. 12-22, 2012.

See Also

`lda`

Examples

```
palette(c("#E41A1C","#377EB8","#4DAF4A"))

# Simulation of data
n = 900; p = 25
n1 = 1/3*n; n2 = 1/3*n; n3 = 1/3*n;
S1 = diag(2)
S2 = rbind(c(1,-0.95),c(-0.95,1))
S3 = rbind(c(2,0),c(0,0.05))
```

```
m1 = c(0,0); m2 = c(0,2); m3 = c(2,0)
X = rbind(mvrnorm(n1,m1,S1),mvrnorm(n2,m2,S2),mvrnorm(n3,m3,S3))
Q = qr.Q(qr(mvrnorm(p,mu=rep(0,p),Sigma=diag(25,p))))
B = mvrnorm(nrow(X),rep(0,p-2),0.1*diag(rep(p-2,p-2)))
X = crossprod(t(cbind(X,B)),Q)
cls = rep(c(1,2,3),c(n1,n2,n3))

# Cross-validation
nbrep = 10
CCR = matrix(NA,2,nbrep)
for (i in 1:nbrep){
  ind = sample(n)[1:(3/5*n)]
  lda.c = lda(X[ind,],cls[ind])
  res = predict(lda.c,X[-ind,])
  CCR[1,i] = sum(res$cl==cls[-ind])/length(cls[-ind])
  prms = pfda(X[ind,],cls[ind],model=c('DkBk','DB','AkB','AB'),crit='bic',display=TRUE)
  res = predict(prms,X[-ind,])
  CCR[2,i] = sum(res$cl==cls[-ind])/length(cls[-ind])
}

# Display results
split.screen(c(2,1))
split.screen(c(1,3), screen = 1)
screen(3)
plot(predict(princomp(X)),col=cls,pch=(17:19)[cls],main='PCA')
screen(4)
plot(crossprod(t(X),lda(X,cls)$scaling),col=cls,pch=(17:19)[cls],main='LDA')
screen(5)
plot(crossprod(t(X),pfda(X,cls,model='DkBk')$V),col=cls,pch=(17:19)[cls],main='PFDA',
  xlab='LD1',ylab='LD2')
screen(2)
boxplot(t(CCR),names=c('LDA','PFDA'),col=c(1,2),ylab="CCR",
  main='CV correct classification rate')
```

predict.pfda — *Prediction method for 'pfda' class objects.*

Description

The prediction method for 'pfda' class objects allows to predict the labels for test observations.

Usage

```
## S3 method for class 'pfda'
predict(object, X, ...)
```

Arguments

object	a supervised classifier genarated by the pfda function (a 'pfda' object).
X	the test data.
...	additional options for internal functions.

Value

A list with:

cls	The predicted class labels.
P	the posterior probabilities that observations belong to the classes.

Author(s)

Charles Bouveyron and Camille Brunet

References

C. Bouveyron and C. Brunet, Probabilistic Fisher discriminant analysis: A robust and flexible alternative to Fisher discriminant analysis, Neurocomputing, vol. 90 (1), pp. 12-22, 2012.

See Also

lda

Examples

```
palette(c("#E41A1C","#377EB8","#4DAF4A"))

# Simulation of data
n = 900; p = 25
n1 = 1/3*n; n2 = 1/3*n; n3 = 1/3*n;
S1 = diag(2)
S2 = rbind(c(1,-0.95),c(-0.95,1))
S3 = rbind(c(2,0),c(0,0.05))
m1 = c(0,0); m2 = c(0,2); m3 = c(2,0)
X = rbind(mvrnorm(n1,m1,S1),mvrnorm(n2,m2,S2),mvrnorm(n3,m3,S3))
Q = qr.Q(qr(mvrnorm(p,mu=rep(0,p),Sigma=diag(25,p))))
B = mvrnorm(nrow(X),rep(0,p-2),0.1*diag(rep(p-2,p-2)))
X = crossprod(t(cbind(X,B)),Q)
cls = rep(c(1,2,3),c(n1,n2,n3))

# Cross-validation
nbrep = 10
CCR = matrix(NA,2,nbrep)
for (i in 1:nbrep){
  ind = sample(n)[1:(3/5*n)]
  lda.c = lda(X[ind,],cls[ind])
  res = predict(lda.c,X[-ind,])
  CCR[1,i] = sum(res$cl==cls[-ind])/length(cls[-ind])
  prms = pfda(X[ind,],cls[ind],model=c('DkBk','DB','AkB','AB'),crit='bic',display=TRUE)
  res = predict(prms,X[-ind,])
  CCR[2,i] = sum(res$cl==cls[-ind])/length(cls[-ind])
}

# Display results
split.screen(c(2,1))
split.screen(c(1,3), screen = 1)
screen(3)
plot(predict(princomp(X)),col=cls,pch=(17:19)[cls],main='PCA')
screen(4)
plot(crossprod(t(X),lda(X,cls)$scaling),col=cls,pch=(17:19)[cls],main='LDA')
screen(5)
plot(crossprod(t(X),pfda(X,cls,model='DkBk')$V),col=cls,pch=(17:19)[cls],main='PFDA',
  xlab='LD1',ylab='LD2')
screen(2)
boxplot(t(CCR),names=c('LDA','PFDA'),col=c(1,2),ylab="CCR",
  main='CV correct classification rate')
```

Index