Package ‘skda’

February 20, 2015

Version 0.1
Date 2013-11-15
Title Sparse (Multicategory) Kernel Discriminant Analysis
Author Len Stefanski, Yichao Wu, and Kyle White
Maintainer Yichao Wu <wu@stat.ncsu.edu>
Depends parallel, R (>= 2.0)
Description Sparse (Multicategory) Kernel Discriminant Analysis does variable selection for nonparametric classification
License GPL-2
Repository CRAN
Date/Publication 2013-12-13 00:34:31
NeedsCompilation yes

R topics documented:

- cvskda .................................................. 1
- predprob ........................................... 3
- skda .................................................. 5

Index 8

---

**cvskda**  
*Cross validation for SKDA*

Description

This function implements cross validation for the sparse (multicategory) kernel discriminant analysis (**skda**).

Usage

```
cvskda(x, y, taus, nfolds=10, method="Bayes")
```
Arguments

- **x**: a matrix ( \( n \times p \) ) that contains predictors.
- **y**: a vector that contains the categorical response coded as 1, 2, ..., \( K \).
- **taus**: a vector that contains the candidate regularization parameters.
- **nfolds**: the number of folds used in cross validation.
- **method**: method (mle or Bayes) to be used in the KDA classifier.

Value

- **lam**: the best solution tuned by cross validation

Author(s)

L. A. Stefanski, Y. Wu, and K. White

References


See Also

- `skda`, `predprob`

Examples

```r
# a binary example
rm(list=ls())
n=200
p=20
r=0.8
x=matrix(rnorm(n*p,mean=0,sd=1),n,p)
y=rbinom(n,1,0.5)
x[,3]=x[,3]+(2*y-1)*r
x[,11]=x[,11]+(2*y-1)*r
y=y+1

ind1=which(y>1.5)
ind0=which(y<1.5)
plot(-4:4, -4:4, type = "n")
points(x[ind1, 3], x[ind1,11],col="blue")
points(x[ind0, 3], x[ind0,11],col="red")
lam=skda(x,y,3)$lam

# for cross validation see demo(cvskda, package="skda", ask=TRUE)
```
# a three-class example
```r
tm(list=ls())
n=200
p=20
r=2
x=matrix(rnorm(n*p,mean=0,sd=1),n,p)
y=ceiling(runif(n,0,3))
theta=c(0, 2*pi/3, 4*pi/3)
x[,3]=x[,3]+r*cos(theta[y])
x[,11]=x[,11]+r*sin(theta[y])

ind1=which(y==1)
ind2=which(y==2)
ind3=which(y==3)

plot(-6:6, -6:6, type = "n")
points(x[ind1, 3], x[ind1,11],col="blue")
points(x[ind2, 3], x[ind2,11],col="red")
points(x[ind3, 3], x[ind3,11],col="black")

lam=skda(x,y,3)$lam
```

# for cross validation see demo(cvsksda, package="skda", ask=TRUE)

### predprob

**Conditional class probability prediction for SKDA**

**Description**
This function predicts the conditional class probability for the sparse (multicategory) kernel discriminant analysis (**skda**).

**Usage**
```
predprob(x, y, lam, xnew, method = "Bayes")
```

**Arguments**
- **x**: a matrix (n X p) that contains predictors.
- **y**: a vector that contains the categorical response coded as 1, 2, ..., K.
- **lam**: a length-K vector of inverse smoothing bandwidths.
- **xnew**: a matrix (with p columns) that contains predictors of new observations.
- **method**: method (mle or Bayes) to be used in the KDA classifier.

**Value**
- **phat**: a matrix (with K columns) that contains predicted conditional class probabilities.
Author(s)

L. A. Stefanski, Y. Wu, and K. White

References


See Also

`skda`, `cvskda`

Examples

```r
# a binary example
rm(list=ls())
n=200
p=20
r=0.8
x=matrix(rnorm(n*p,mean=0,sd=1),n,p)
y=rbinom(n,1,0.5)
x[,3]=x[,3]+(2*y-1)*r
x[,11]=x[,11]+(2*y-1)*r

y=y+1
ind1=which(y>1.5)
ind0=which(y<1.5)
plot(-4:4, -4:4, type = "n")
points(x[ind1, 3], x[ind1,11],col="blue")
points(x[ind0, 3], x[ind0,11],col="red")

lam=skda(x,y,3)$lam

xtest=matrix(rnorm(10*n*p,mean=0,sd=1),10*n,p)
ytest=rbinom(10*n,1,0.5)
xtest[,3]=xtest[,3]+(2*ytest-1)*r
xtest[,11]=xtest[,11]+(2*ytest-1)*r
ytest=ytest+1

ptesthat=predprob(x,y,lam, xtest)
ytesthat=apply(ptesthat, 1,which.max)
error=sum(abs(ytesthat-ytest)>0.5)
error/length(ytest)
```
# a three-class example
```
rm(list=ls())
n=200
p=20
r=2
x=matrix(rnorm(n*p,mean=0,sd=1),n,p)
y=ceiling(runif(n,0,3))
thes=c(0, 2*pi/3, 4*pi/3)
x[,3]=x[,3]+r*cos(thetas[y])
x[,11]=x[,11]+r*sin(thetas[y])

ind1=which(y==1)
ind2=which(y==2)
ind3=which(y==3)
```
```
plot(-6:6, -6:6, type = "n")
points(x[ind1, 3], x[ind1,11],col="blue")
points(x[ind2, 3], x[ind2,11],col="red")
points(x[ind3, 3], x[ind3,11],col="black")
```
```
lam=skda(x,y,3)$lam
```
```
xtest=matrix(rnorm(10*n*p,mean=0,sd=1),10*n,p)
ytest=ceiling(runif(10*n,0,3))
xtest[,3]=xtest[,3]+r*cos(thetas[ytest])
xtest[,11]=xtest[,11]+r*sin(thetas[ytest])
```
```
ptesthat=predprob(x,y,lam, xtest)
ytesthat=apply(ptesthat, 1,which.max)
error=sum(abs(ytesthat-ytest)>0.5)
error/length(ytest)
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
Usage

```
skda(x, y, tau, method="Bayes")
```

Arguments

- **x**: a matrix \((n \times p)\) that contains predictors.
- **y**: a vector that contains the categorical response coded as 1, 2, ..., \(K\).
- **tau**: a positive number that is the regularization parameter.
- **method**: method (mle or Bayes) to be used in the KDA classifier.

Value

- **lam**: the SKDA solution of size \(p \times 1\).
- **phat**: the predicted conditional class probabilities of size \(n \times K\).

Author(s)

L. A. Stefanski, Y. Wu, and K. White

References


See Also

```
predprob, cvskda
```

Examples

```r
# a binary example
rm(list=ls())
n=200
p=20
r=0.8
x=matrix(rnorm(n*p,mean=0,sd=1),n,p)
y=rbinom(n,1,0.5)
x[,3]=x[,3]+(2*y-1)*r
x[,11]=x[,11]+(2*y-1)*r

y=y+1

ind1=which(y>1.5)
ind0=which(y<1.5)
plot(-4:4, -4:4, type = "n")
```
# a three-class example
rm(list=ls())
n=200
p=20
r=2
x=matrix(rnorm(n*p,mean=0,sd=1),n,p)
y=ceiling(runif(n,0,3))
thes=c(0, 2*pi/3, 4*pi/3)
x[,3]=x[,3]+r*cos(thes[y])
x[,11]=x[,11]+r*sin(thes[y])

ind1=which(y==1)
ind2=which(y==2)
ind3=which(y==3)

plot(-6:6, -6:6, type = "n")
points(x[ind1, 3], x[ind1,11],col="blue")
points(x[ind2, 3], x[ind2,11],col="red")
points(x[ind3, 3], x[ind3,11],col="black")
lam=skda(x,y,3)$lam
Index

*Topic file
  cvsksda, 1
  predprob, 3
  skda, 5

cvskda, 1, 4–6
predprob, 2, 3, 5, 6
skda, 1–5, 5