

Package ‘bigdatadist’

May 7, 2026

Version 1.1

Date 2018-09-23

Title Distances for Machine Learning and Statistics in the Context of Big Data

Description Functions to compute distances between probability measures or any other data object than can be posed in this way, entropy measures for samples of curves, distances and depth measures for functional data, and the Generalized Mahalanobis Kernel distance for high dimensional data. For further details about the metrics please refer to Martos et al (2014) <[doi:10.3233/IDA-140706](https://doi.org/10.3233/IDA-140706)>; Martos et al (2018) <[doi:10.3390/e20010033](https://doi.org/10.3390/e20010033)>; Hernandez et al (2018, submitted); Martos et al (2018, submitted).

Author Gabriel Martos [aut, cre],
Nicolas Hernandez [aut]

Depends R (>= 3.4.0)

Maintainer Gabriel Martos <gmartos@utdt.edu>

License GPL (>= 3)

Repository CRAN

Imports MASS, FNN, rrcov, pdist

NeedsCompilation yes

Date/Publication 2018-09-24 13:30:03 UTC

Contents

Ausmale	2
entropy	2
entropy.fd	3
fdframe	4
gmdepth	5
gmdepth.fd	6
kmdepth.fd	7
levelsetdist	8
merval	9
rkhs	9

Index**11**

Ausmale	<i>Australian Male Mortality Rates</i>
---------	--

Description

The data consist of set of measurements across years of male mortality rates in Australia from package `fds`.

Usage

```
Ausmale
```

Format

A list with years in the first component and a 101 times 103 matrix, years in rows and cohorts in columns, in the second component.

Source

```
fds
```

entropy	<i>Entropy Computation</i>
---------	----------------------------

Description

This function allows you to compute the family of alpha entropy as stated in Martos et al (2018).

Usage

```
entropy(X,alpha=2,k.neighbor,scale=FALSE)
```

Arguments

<code>X</code>	data in a matrix where variables are in columns and observations are in rows.
<code>alpha</code>	a parameter defining the entropy function.
<code>k.neighbor</code>	number of neighbour points to consider in the computation of entropy.
<code>scale</code>	logical variable indicating if scaling is required.

Details

The function computes the alpha entropy and the local alpha entropy (see reference for further details) of a data set using a non parametric density estimator.

Value

local.entropy local entropy relative to each point in the sample.
entropy estimated entropy.

References

Martos, G. et al (2018): Entropy Measures for Stochastic Processes with Applications in Functional Anomaly Detection. Entropy 20(1): 33 (2018).

Examples

```
require(MASS);
data = mvrnorm(100,c(0,0),diag(2));
entropy(data, alpha = 2, k.neighbor = 10, scale = FALSE)
```

entropy.fd *Functional Entropy Measures*

Description

This function allows you to compute the family of alpha-Entropy for functional data as stated in Martos et al (2018).

Usage

```
entropy.fd(fdframe, gamma = 1, kerfunc="rbf",
           kerpar = list(sigma = 1, bias=0,degree=2),
           alpha=2,d=2,resol,k.neighbor)
```

Arguments

fdframe functional data frame fdframe object.
gamma regularization parameter.
kerfunc kernel function (rbf or poly) to be used.
kerpar a list of kernel parameters where sigma is the scale with both kernels.
alpha Entropy parameter.
d Dimension truncation in the Reproducing Kernel Hilbert Space representation.
resol number of level sets used to compute the functional entropy.
k.neighbor number of points to estimate the support of the distribution.

Details

This function estimates the entropy of a stochastic process. To this aim, the raw functional data is projected onto a Reproducing Kernel Hilbert Space, and the entropy is estimated using the coefficient of these functions.

Value

local.entropy local entropy relative to each curve in the sample.
entropy estimated entropy of the the set of functions.

Author(s)

Hernandez and Martos

References

Martos, G. et al (2018). Entropy Measures for Stochastic Processes with Applications in Functional Anomaly Detection. Entropy 20(1), 33 (2018).

Examples

```
data(Ausmale); t <- Ausmale[[1]]
t <- as.numeric(( t - min(t) ) / length(t))
raw.data <- fdframe(t=t, Y=Ausmale[[2]])

entropy.fd(raw.data, gamma=0.0001, kerfunc="rbf", kerpar=c(10),
           alpha=2, k.neighbor=15)
```

fdframe

Functional Data Frame

Description

This function is used to create multivariate functional data frame objects to be used in combination with the functions in the package bigdatadist.

Usage

```
fdframe(t, Y)
```

Arguments

t abscissa values at which observations took place.
Y matrix with functions in columns and observations in rows.

Examples

```
t = 1:10; Y = cbind(sin(t),cos(t))
fdata = fdframe(t,Y)
plot(fdata, xlab='Time', ylab='')
```

Description

This function allows you to compute the Generalized Kernel Mahalanobis depth measure as stated in Hernandez et al (2018, submitted) and the Generalized Mahalanobis distance in Martos et al (2014).

Usage

```
gmdepth(A,b,resol,k.neighbor)
```

Arguments

A	data matrix where variables in columns, observations in rows.
b	a new point in the support of the distribution to evaluate the depth. If omitted, the function compute the distances and depth between all points in the sample.
resol	resolution level, i.e. number of density level sets to consider.
k.neighbor	number of local neighbours to estimate the support.

Value

depth	the generalized Mahalanobis depth measure.
distance	the generalized Mahalanobis distance measure.

Author(s)

Hernandez and Martos

References

Hernandez N. et al (2018). Generalized Mahalanobis depth functions (submitted). Martos, G. et al (2014). Generalizing the Mahalanobis distance via density kernels. Intelligent Data Anal.

Examples

```
require(MASS)
set.seed(1)
A=mvnrnorm(450,c(0,0),Sigma=diag(2))
b=mvnrnorm(50,c(10,10),Sigma=diag(c(0.1,0.1)))
C=rbind(A,b)
plot(C, pch=20, col=c(rep('black',450),rep('red',50)),
      xlab='x1',ylab='x2')

gmd.fit = gmdepth(A=C)
depth   = gmd.fit$depth
```

```

distance = gmd.fit$distance
plot(depth,distance, pch=20,
      col=c(rep('black',450),rep('red',50)))
gmdepth(A=A,b=mvrnorm(1,c(0,0),Sigma=diag(2)))

```

gmdepth.fd	<i>Generalized Mahalanobis Kernel Depth and Distance for Functional Data</i>
------------	--

Description

This function allows you to compute the Generalized Kernel Mahalanobis depth measure as stated in Hernandez et al (2018, submitted) and the Generalized Mahalanobis distance in Martos et al (2014), for functional data represented in a Reproducing Kernel Hilbert Space.

Usage

```

gmdepth.fd(fdframe, gamma = 1,kerfunc="rbf" ,
           kerpar=list(sigma=1,bias=0,degree=2),d=2,resol,k.neighbor)

```

Arguments

fdframe	an fdframe object storing raw functional data.
gamma	regularization parameter.
kerfunc	kernel function to be used.
kerpar	a list of kernel parameters where sigma is the scale with both kernels.
d	truncation parameter in the Reproducing Kernel Hilbert Space representation.
resol	resolution level to estimate the generalized Mahalanobis distance.
k.neighbor	number of neighbours to estimate the support of the distribution.

Value

depth	the generalized Mahalanobis depth measure for the curves in the sample.
distance	the generalized Mahalanobis distance for the curves in the sample.

Author(s)

Hernandez and Martos

References

Hernandez N. et al (2018, submitted). Generalized Mahalanobis depth functions. Martos, G. et al (2014). Generalizing the Mahalanobis distance via density kernels. Intelligent Data Anal.

Examples

```

data(Ausmale); t <- Ausmale[[1]]
t = as.numeric(( t - min(t) ) / length(t))
raw.data = fdframe(t=t, Y=Ausmale[[2]])

gmd.fit.fd = gmdepth.fd(raw.data,gamma=0.001,
                        kerfunc="rbf",kerpar=list(sigma = 10))

gmd.fit.fd$distance
gmd.fit.fd$depth

rbPal <- colorRampPalette(c('red','black'))
color = rbPal(5)[as.numeric(cut(gmd.fit.fd$depth,breaks = 5))]
plot(rkhs(raw.data,gamma=0.0001,kerfunc="rbf",kerpar=list(sigma = 10)),
     col = color, xlab='time',ylab='')

```

kmdepth.fd

*Kernel Mahalanobis Depth for Functional Data***Description**

This function allows you to compute the Generalized Kernel Mahalanobis depth measure for a sample of functional data as stated in Hernandez et al (2018, submitted).

Usage

```

kmdepth.fd(fdframe, gamma = 1, kerfunc = "rbf" ,
            kerpar = list(sigma = 1, bias = 0, degree = 2) ,
            d = 2 , robust=TRUE , h=0.1 , nsamp=250)

```

Arguments

fdframe	an fdframe object storing raw functional data.
gamma	regularization parameter.
kerfunc	kernel function to be used.
kerpar	a list of kernel parameters where sigma is the scale with both kernels.
d	truncation parameter in the Reproducing Kernel Hilbert Space representation.
robust	TRUE if the covariance matrix is estimated through Robust Maximum Likelihood method.
h	numeric parameter controlling the a-priori percentage of outliers in the sample (value between 0 and 1, by def = 0.1).
nsamp	number of subsets used for initial estimates (by def = 250).

Value

depth	the kernel-mahalanobis depth measure for the curves in the sample.
-------	--

Author(s)

Hernandez and Martos

References

Hernandez N. et al (2018, submitted). Generalized Mahalanobis depth functions.

Examples

```
data(Ausmale); t <- Ausmale[[1]]
t = as.numeric(( t - min(t) ) / length(t))
raw.data = fdframe(t=t, Y=Ausmale[[2]])

kmd.fit.fd = kmdepth.fd(raw.data, gamma = 0.0001, kerfunc = "rbf" ,
                        kerpar = list(sigma = 10) , d = 2 , robust=TRUE)

kmd.fit.fd$depth

rbPal <- colorRampPalette(c('red', 'black'))
color = rbPal(5)[as.numeric(cut(kmd.fit.fd$depth, breaks = 5))]
plot(rkhs(raw.data, gamma=0.0001, kerfunc="rbf", kerpar=list(sigma = 10)),
     col = color, xlab='time', ylab='')
```

levelsetdist

Level Set Distances

Description

This function allows you to compute the alpha level set distances as proposed in Martos et al. Nonparametric distances for probability measures with applications, 2018 (submitted).

Usage

```
levelsetdist(A,B,n.level=10,k.neighbor=10)
```

Arguments

A	data set in a matrix where variables are in columns and observations are in rows.
B	data set in a matrix where variables are in columns and observations are in rows.
n.level	the number of level sets to consider for distance computation.
k.neighbor	number of neighbour points to consider in the estimation of the support of the distribution on each class.

Details

The function computes the alpha level set distance between two (samples from) different probability measures. Details about the distance and the criterion to fix its hyperparameter can be found in Martos et al (2018, submitted).

Value

distance distance estimation between the two data sets or distributions.

References

Martos, G. et al (2018): Nonparametric distances for probability measures with applications in classification. J. of Classification, 2018 (submitted).

Examples

```
require(MASS);
set.seed(1)
A = mvrnorm(100,c(0,0),diag(2)); B = mvrnorm(150,c(1,1),diag(2));
levelsetdist(A, B)
```

merval

Merval Index

Description

The data consist of an low and high values of the Merval Index stock market from Argentina; the data were gathered from Yahoo Finance.

Usage

```
merval
```

Format

A dataframe with 5269 observations with daily minimum, maximum, open and close index values.

Source

Yahoo Finance

rkhs

RKHS Representation

Description

This function allows you to fit discrete functional data (fdframe) as functions in RKHS solving a regularization problem as stated in Munoz (2010).

Usage

```
rkhs(fdframe, gamma=1, kerfunc='rbf',
      kerpar=list(sigma=1, bias=0, degree=2))
```

Arguments

fdframe	functional data fdframe object.
gamma	regularization parameter.
kerfunc	kernel function rbf or poly to be used.
kerpar	a list of kernel parameters where sigma is the scale with both kernels.

Value

fdframe	raw data in an fdframe object.
f	estimated functional data
alpha	coefficients for the linear combination.
lambda.star	reduced coefficients for the linear combination.

Author(s)

Hernandez and Martos

References

Munoz A. et al (2010). Representing functional data using support vector machines. Pattern recognition letters, 31(6).

Examples

```
data(merval); t <- as.Date(merval[1:100,1])
t <- as.numeric(( t - min(t) ) / 154)
raw.data <- fdframe(t = t, Y = merval[1:100,2:5])
plot(raw.data, xlab='time', ylab='Merval raw data')

f.data <- rkhs(raw.data, gamma = 0.001, kerpar = list(sigma = 8))

print(f.data)

plot(f.data, xlab='time', ylab='Merval data')
```

Index

- * **Entropy**
 - entropy, 2
 - * **Generalized Mahalanobis depth and distance.**
 - gmdepth, 5
 - * **Kernel depth**
 - gmdepth.fd, 6
 - kmdepth.fd, 7
 - * **Probability metrics**
 - levelsetdist, 8
 - * **datasets**
 - Ausmale, 2
 - merval, 9
 - * **multivariate time series formatting.**
 - fdframe, 4
 - * **rkhs, Entropy**
 - entropy.fd, 3
 - * **rkhs**
 - rkhs, 9
- Ausmale, 2
- entropy, 2
entropy.fd, 3
- fdframe, 4
- gmdepth, 5
gmdepth.fd, 6
- kmdepth.fd, 7
- levelsetdist, 8
- merval, 9
- rkhs, 9