Package ‘lfda’

July 31, 2019

Type Package
Title Local Fisher Discriminant Analysis
Version 1.1.3
Date 2019-07-31

URL https://github.com/terrytangyuan/lfda

BugReports https://github.com/terrytangyuan/lfda/issues

Maintainer Yuan Tang <terrytangyuan@gmail.com>
License MIT + file LICENSE

Description Functions for performing and visualizing Local Fisher Discriminant Analysis (LFDA), Kernel Fisher Discriminant Analysis (KLFDA), and Semi-supervised Local Fisher Discriminant Analysis (SELF).

Depends R (>= 3.1.0)
Imports plyr, grDevices, rARPACK
Suggests testthat, rgl

RoxygenNote 6.1.0

NeedsCompilation no

Author Yuan Tang [aut, cre] (<https://orcid.org/0000-0001-5243-233X>),
Wenxuan Li [ctb],
Nan Xiao [ctb, cph],
Zachary Deane-Mayer [ctb]

Repository CRAN

Date/Publication 2019-07-31 17:10:02 UTC

R topics documented:

Cols ................................................................. 2
getAffinityMatrix .............................................. 2
getMetricOfType ................................................ 3
klfda .......................................................... 3
kmatrixGauss .................................................. 4
getAffinityMatrix

Description
This function returns an affinity matrix within knn-nearest neighbors from the distance matrix.

Usage
getAffinityMatrix(distance2, knn, nc)

Arguments
- distance2: The distance matrix for each observation
- knn: The number of nearest neighbors
- nc: The number of observations for data in this class

Value
an affinity matrix - the larger the element in the matrix, the closer two data points are
getMetricOfType

Description
This function returns the requested type of transforming metric.

Usage
getMetricOfType(metric, eigVec, eigVal, total)

Arguments
- metric: The type of metric to be requested
- eigVec: The eigenvectors of the problem
- eigVal: The eigenvalues of the problem
- total: The number of total rows to be used for weighting denominator

Value
The transformation metric in requested type

klfda

Kernel Local Fisher Discriminant Analysis for Supervised Dimensionality Reduction

Description
Performs kernel local fisher discriminant analysis on the given data, which is the non-linear version of LFDA (see details lFDA).

Usage
klfda(k, y, r, metric = c("weighted", "orthonormalized", "plain"),
      knn = 6, reg = 0.001)

Arguments
- k: n x n kernel matrix. Result of the kmatrixGauss function. n is the number of samples
- y: n dimensional vector of class labels
- r: dimensionality of reduced space (default: d)
- metric: type of metric in the embedding space (default: 'weighted') 'weighted' — weighted eigenvectors 'orthonormalized' — orthonormalized 'plain' — raw eigenvectors
- knn: parameter used in local scaling method (default: 6)
- reg: regularization parameter (default: 0.001)
kmatrixGauss

**Value**

- d x r transformation matrix \((Z = t(T) \ast X)\)
- r x n matrix of dimensionality reduced samples

**Author(s)**

Yuan Tang

**References**


Original Matlab Implementation: http://www.ms.k.u-tokyo.ac.jp/software.html#LFDA

**See Also**

See `lfda` for the linear version.

**Examples**

```r
k <- kmatrixGauss(iris[, -5])
y <- iris[, 5]
r <- 3
kldfa(k, y, r, metric = "plain")
```

---

<table>
<thead>
<tr>
<th>kmatrixGauss</th>
<th>Gaussian Kernel Computation (Particularly used in Kernel Local Fisher Discriminant Analysis)</th>
</tr>
</thead>
</table>

**Description**

Gaussian kernel computation for klfd, which maps the original data space to non-linear and higher dimensions.

**Usage**

```r
kmatrixGauss(x, sigma = 1)
```
Arguments

- `x`: A n x d matrix of original samples. `n` is the number of samples.
- `sigma`: Dimensionality of reduced space. (default: 1)

Value

- `K`: A n x n kernel matrix. `n` is the number of samples.

Author(s)

- Yuan Tang

References


See Also

- See `klfda` for the computation of kernel local fisher discriminant analysis

Examples

```r
kmatrixGauss(iris[, -5])
```

---

`lfda` 

**Local Fisher Discriminant Analysis for Supervised Dimensionality Reduction**

**Description**

Performs local fisher discriminant analysis (LFDA) on the given data.

**Usage**

```r
lfda(x, y, r, metric = c("orthonormalized", "plain", "weighted"),
    knn = 5)
```
Arguments

- **x**: n x d matrix of original samples. n is the number of samples.
- **y**: length n vector of class labels
- **r**: dimensionality of reduced space (default: d)
- **metric**: type of metric in the embedding space (no default) 'weighted' — weighted eigenvectors 'orthonormalized' — orthonormalized 'plain' — raw eigenvectors
- **knn**: parameter used in local scaling method (default: 5)

Details

LFDA is a method for linear dimensionality reduction that maximizes between-class scatter and minimizes within-class scatter while at the same time maintain the local structure of the data so that multimodal data can be embedded appropriately. Its limitation is that it only looks for linear boundaries between clusters. In this case, a non-linear version called kernel LFDA will be used instead. Three metric types can be used if needed.

Value

- list of the LFDA results:
  - **T**: d x r transformation matrix (Z = x * T)
  - **Z**: n x r matrix of dimensionality reduced samples

Author(s)

Yuan Tang

References


See Also

See klfda for the kernelized variant of LFDA (Kernel LFDA).

Examples

```r
k <- iris[, -5]
y <- iris[, 5]
r <- 3
lfda(k, y, r, metric = "plain")
```
plot.lfda

3D Visualization for LFDA/KLFDA Result

Description
This function plot 3 dimensions of the lfda/klfda result.

Usage
```r
## S3 method for class 'lfda'
plot(x, labels, cleanText = FALSE, ...)
```

Arguments
- `x`: The lfda/klfda result.
- `labels`: A list of class labels used for lfda/klfda training.
- `cleanText`: A boolean value to specify whether to make the labels in the plot cleaner (default: FALSE)
- `...`: Additional arguments

See Also
See `lfda` and `klfda` for the metric learning method used for this visualization.

predict.lfda

LFDA Transformation/Prediction on New Data

Description
This function transforms a data set, usually a testing set, using the trained LFDA metric

Usage
```r
## S3 method for class 'lfda'
predict(object, newdata = NULL, type = "raw", ...)
```

Arguments
- `object`: The result from lfda function, which contains a transformed data and a transforming matrix that can be used for transforming testing set
- `newdata`: The data to be transformed
- `type`: The output type, in this case it defaults to "raw" since the output is a matrix
- `...`: Additional arguments
Value

the transformed matrix

Author(s)

Yuan Tang

Examples

```r
k <- iris[, -5]
y <- iris[, 5]
r <- 3
model <- lfda(k, y, r = 4, metric = "plain")
predict(model, iris[, -5])
```

---

**print.lfda**

*Print an lfda object*

Description

Print an lfda object

Usage

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

Arguments

- **x**
  
  The result from lfda function, which contains a transformed data and a transforming

- **...**
  
  ignored

---

**repmat**

*Matlab-Syntaxed Repmat*

Description

This function mimics the behavior and syntax of repmat() in Matlab it generates a large matrix consisting of an N-by-M tiling copies of A

Usage

```r
repmat(A, N, M)
```
Arguments

A  
original matrix to be used as copies

N  
the number of rows of tiling copies of A

M  
the number of columns of tiling copies of A

Value

matrix consisting of an N-by-M tiling copies of A

Description

Performs semi-supervised local fisher discriminant analysis (SELF) on the given data. SELF is a linear semi-supervised dimensionality reduction method smoothly bridges supervised LFDA and unsupervised principal component analysis, by which a natural regularization effect can be obtained when only a small number of labeled samples are available.

Usage

self(X, Y, beta = 0.5, r, metric = c("orthonormalized", "plain", "weighted"), kNN = 5, minObsPerLabel = 5)

Arguments

X  
n x d matrix of original samples. n is the number of samples.

Y  
length n vector of class labels

beta  
degree of semi-supervisedness (0 <= beta <= 1; default is 0.5 ) 0: totally supervised (discard all unlabeled samples) 1: totally unsupervised (discard all label information)

r  
dimensionality of reduced space (default: d)

metric  
type of metric in the embedding space (no default) 'weighted' — weighted eigenvectors 'orthonormalized' — orthonormalized 'plain' — raw eigenvectors

kNN  
parameter used in local scaling method (default: 5)

minObsPerLabel  
the minimum number observations required for each different label (default: 5)

Value

list of the SELF results:

T  
d x r transformation matrix (Z = x * T)

Z  
n x r matrix of dimensionality reduced samples
Author(s)
Yuan Tang

References

See Also
See `lfda` for LFDA and `klfda` for the kernelized variant of LFDA (Kernel LFDA).

Examples
```r
x <- iris[, -5]
y <- iris[, 5]
self(x, y, beta = 0.1, r = 3, metric = "plain")
```

Description
This function defines operation for negative one half matrix power operator

Usage
```r
x %^% n
```

Arguments
```r
x the matrix we want to operate on
n the exponent
```

Value
the matrix after negative one half power

%-- Negative One Half Matrix Power Operator

%--

%--

%``
Index

*Topic discriminant
  klfda, 3
  kmatrixGauss, 4
  lfda, 5
  self, 9

*Topic fisher
  klfda, 3
  kmatrixGauss, 4
  lfda, 5
  self, 9

*Topic kernel
  kmatrixGauss, 4

*Topic klfda
  klfda, 3
  kmatrixGauss, 4

*Topic lfda
  lfda, 5
  self, 9

*Topic local
  klfda, 3
  kmatrixGauss, 4
  lfda, 5
  self, 9

*Topic mahalanobis
  klfda, 3
  kmatrixGauss, 4
  lfda, 5
  self, 9

*Topic metric
  klfda, 3
  kmatrixGauss, 4
  lfda, 5
  self, 9

*Topic semi-supervised
  self, 9

*Topic transformation
  klfda, 3
  kmatrixGauss, 4
  lfda, 5

self, 9
%^%, 10

Cols, 2

getAffinityMatrix, 2
getMetricOfType, 3
klfda, 3, 6, 7, 10
kmatrixGauss, 3, 4

lfda, 3, 4, 5, 7, 10

plot.lfda, 7
predict.lfda, 7
print.lfda, 8

repmat, 8

self, 9