

Package ‘RECA’

May 16, 2018

Type Package

Title Relevant Component Analysis for Supervised Distance Metric Learning

Version 1.6

Maintainer Nan Xiao <me@nanx.me>

Description Relevant Component Analysis (RCA) tries to find a linear transformation of the feature space such that the effect of irrelevant variability is reduced in the transformed space.

License GPL-3 | file LICENSE

URL <https://nanx.me/RECA/>, <https://github.com/road2stat/RECA>

BugReports <https://github.com/road2stat/RECA/issues>

Suggests MASS

Encoding UTF-8

RoxygenNote 6.0.1.9000

NeedsCompilation no

Author Nan Xiao [aut, cre] (<<https://orcid.org/0000-0002-0250-5673>>)

Repository CRAN

Date/Publication 2018-05-16 18:29:31 UTC

R topics documented:

RECA-package	2
rca	2
Index	5

 RECA-package

Relevant Component Analysis

Description

The RECA package offers an R implementation for Relevant Component Analysis (RCA).

Details

Package: RECA
 Type: Package
 License: GPL-3

Note

Bug reports and feature requests can be sent to <https://github.com/road2stat/RECA/issues>.

 rca

Relevant Component Analysis

Description

`rca` performs relevant component analysis (RCA) for the given data. It takes a data set and a set of positive constraints as arguments and returns a linear transformation of the data space into better representation, alternatively, a Mahalanobis metric over the data space.

The new representation is known to be optimal in an information theoretic sense under a constraint of keeping equivalent data points close to each other.

Usage

```
rca(x, chunks, useD = NULL)
```

Arguments

<code>x</code>	<code>n * d</code> matrix or data frame of original data.
<code>chunks</code>	a vector of size <code>N</code> describing the chunklets: <code>-1</code> in the <code>i</code> -th place says that point <code>i</code> does not belong to any chunklet; integer <code>j</code> in place <code>i</code> says that point <code>i</code> belongs to chunklet <code>j</code> ; The chunklets indexes should be <code>1:number-of-chunklets</code> .
<code>useD</code>	optional. When not given, RCA is done in the original dimension and <code>B</code> is full rank. When <code>useD</code> is given, RCA is preceded by constraints based LDA which reduces the dimension to <code>useD</code> . <code>B</code> in this case is of rank <code>useD</code> .

Details

The three returned objects are just different forms of the same output. If one is interested in a Mahalanobis metric over the original data space, the first argument is all she/he needs. If a transformation into another space (where one can use the Euclidean metric) is preferred, the second returned argument is sufficient. Using A and B are equivalent in the following sense:

if $y_1 = A * x_1, y_2 = A * x_2$ then

$$(x_2 - x_1)^T * B * (x_2 - x_1) = (y_2 - y_1)^T * (y_2 - y_1)$$

Value

A list of the RCA results:

- B: The RCA suggested Mahalanobis matrix. Distances between data points x_1, x_2 should be computed by $(x_2 - x_1)^T * B * (x_2 - x_1)$
- RCA: The RCA suggested transformation of the data. The data should be transformed by $RCA * data$
- newX: The data after the RCA transformation. $newX = data * RCA$

Note

Note that any different sets of instances (chunklets), e.g. {1, 3, 7} and {4, 6}, might belong to the same class and might belong to different classes.

Author(s)

Nan Xiao <<https://nanx.me>>

References

Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall (2003). Learning Distance Functions using Equivalence Relations. *Proceedings of 20th International Conference on Machine Learning (ICML2003)*

Examples

```
library("MASS") # generate synthetic multivariate normal data
set.seed(42)
k <- 100L # sample size of each class
n <- 3L # specify how many classes
N <- k * n # total sample size
x1 <- mvrnorm(k, mu = c(-16, 8), matrix(c(15, 1, 2, 10), ncol = 2))
x2 <- mvrnorm(k, mu = c(0, 0), matrix(c(15, 1, 2, 10), ncol = 2))
x3 <- mvrnorm(k, mu = c(16, -8), matrix(c(15, 1, 2, 10), ncol = 2))
x <- as.data.frame(rbind(x1, x2, x3)) # predictors
y <- gl(n, k) # response

# fully labeled data set with 3 classes
# need to use a line in 2D to classify
plot(x[, 1L], x[, 2L],
```

```

    bg = c("#E41A1C", "#377EB8", "#4DAF4A")[y],
    pch = rep(c(22, 21, 25), each = k)
  )
  abline(a = -10, b = 1, lty = 2)
  abline(a = 12, b = 1, lty = 2)

  # generate synthetic chunklets
  chunks <- vector("list", 300)
  for (i in 1:100) chunks[[i]] <- sample(1L:100L, 10L)
  for (i in 101:200) chunks[[i]] <- sample(101L:200L, 10L)
  for (i in 201:300) chunks[[i]] <- sample(201L:300L, 10L)

  chks <- x[unlist(chunks), ]

  # make "chunklet" vector to feed the chunks argument
  chunksvec <- rep(-1L, nrow(x))
  for (i in 1L:length(chunks)) {
    for (j in 1L:length(chunks[[i]])) {
      chunksvec[chunks[[i]][j]] <- i
    }
  }

  # relevant component analysis
  rcs <- rca(x, chunksvec)

  # learned transformation of the data
  rcs$RCA

  # learned Mahalanobis distance metric
  rcs$B

  # whitening transformation applied to the chunklets
  chkTransformed <- as.matrix(chks) %*% rcs$RCA

  # original data after applying RCA transformation
  # easier to classify - using only horizontal lines
  xnew <- rcs$newX
  plot(xnew[, 1L], xnew[, 2L],
       bg = c("#E41A1C", "#377EB8", "#4DAF4A")[gl(n, k)],
       pch = c(rep(22, k), rep(21, k), rep(25, k))
  )
  abline(a = -15, b = 0, lty = 2)
  abline(a = 16, b = 0, lty = 2)

```

Index

rca, [2](#)
RECA-package, [2](#)