Package ‘hsicCCA’

February 20, 2015

Type Package

Title Canonical Correlation Analysis based on Kernel Independence Measures

Version 1.0

Date 2013-03-13

Author Billy Chang

Maintainer Billy Chang <billy.chang@mail.utoronto.ca>

Description Canonical correlation analysis that extracts nonlinear correlation through the use of Hilbert Schmidt Independence Criterion and Centered Kernel Target Alignment.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2013-03-14 07:21:57

\r topics documented:

hsicCCA-package ................................................................. 2
hsicCCA .......................................................... 2
hsicCCAfunc ...................................................... 4
ktaCCA .......................................................... 6
ktaCCAfunc .................................................... 7
sumWtDiff ...................................................... 9

Index 10
hsicCCA-package

Canonical Correlation Analysis based on Kernel Independence Measures

Description
Canonical correlation analysis that extracts nonlinear correlation through the use of Hilbert Schmidt Independence Criterion and Centered Kernel Target Alignment.

Details

<table>
<thead>
<tr>
<th>Package:</th>
<th>hsicCCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>1.0</td>
</tr>
<tr>
<td>Date:</td>
<td>2013-03-13</td>
</tr>
<tr>
<td>License:</td>
<td>GPL-2</td>
</tr>
</tbody>
</table>

Author(s)

Billy Chang: <billy.chang@mail.utoronto.ca>

References


hsicCCA

Canonical Correlation Analysis based on the Hilbert-Schmidt Independence Criterion.

Description
Given two multi-dimensional data sets, find pairs of canonical projection pairs that maximize the HSIC criterion.

Usage

hsicCCA(x, y, M, sigmax = NULL, sigmay = NULL, numrepeat = 5, numiter = 100, reltolstop = 1e-04)
Arguments

- **x**: The x-variable data matrix. One row per observation.
- **y**: The y-variable data matrix. One row per observation.
- **M**: Number of canonical projection pairs to extract.
- **sigmax**: The bandwidth parameter for the Gaussian kernel on the x-variable set. A positive value. The smaller the smoother. If NULL, set to median(dist(x)), and will be updated automatically for extracting different pairs of canonical projection.
- **sigmay**: The bandwidth parameter for the Gaussian kernel on the y-variable set. A positive value. The smaller the smoother. If NULL, set to median(dist(y)), and will be updated automatically for extracting different pairs of canonical projection.
- **numrepeat**: Number of random restarts.
- **numiter**: Maximum number of iterations for extracting each pair of canonical projections.
- **reltolstop**: Convergence threshold. Algorithm stops when relative change in cost from consecutive iterations is less than the threshold and will then move on to find the next pair of canonical vectors.

Details

Optimization is done by gradient descent, where Nelder-Mead is used for step-size selection. Nelder Mead may fail to increase the cost at times (when stuck at local minima). User may consider restarting the algorithm when this happens.

Value

A list containing:

- **wx**: The M canonical projection vectors for the x-variable set. Each column corresponds to a projection vector.
- **wy**: The M canonical projection vectors for the y-variable set. Each column corresponds to a projection vector.

Note

Current implementation is slow and requires high storage for large sample data. Sample size > 2000 not recommended.

Author(s)

Billy Chang

References


See Also

    ktaCCA, hsicCCAfunc

Examples

    set.seed(1)
    numData <- 100
    numDim <- 3
    x <- matrix(rnorm(numData*numDim),numData,numDim)
    y <- matrix(rnorm(numData*numDim),numData,numDim)
    z <- runif(numData,-pi,pi)
    y[,1] <- cos(z)+rnorm(numData,sd=0.1); x[,1] <- sin(z)+rnorm(numData,sd=0.1)
    y[,2] <- x[,2]+rnorm(numData,sd=0.5)
    x <- scale(x)
    y <- scale(y)
    fit <- hsicCCA(x,y,2,numrepeat=2,numiter=10)
    par(mfrow=c(1,2))
    for (k in 1:2) plot(x%*%fit$wx[,k],y%*%fit$wy[,k])

hsicCCAfunc    Canonical Correlation Analysis based on the Hilbert-Schmidt Independence Criterion.

Description

    Given two multi-dimensional data sets, find a pair of canonical projection pairs that maximizes the
    HSIC criterion. Called by hsicCCA, and intended for internal use, but users may play with it for
    potential finer controls.

Usage

    hsicCCAfunc(x, y, Wx = NULL, Wy = NULL, sigmax, sigmay, numiter = 20, reltolstop = 1e-04)

Arguments

    x    The x-variable data set. One row per observation.
    y    The y-variable data set. One row per observation.
    Wx   Initial projection vector for the x data set. Randomly set if NULL.
    Wy   Initial projection vector for the y data set. Randomly set if NULL.
    sigmax The bandwidth parameter for the Gaussian kernel on the x-variable set. A positive value. The smaller the smoother.
    sigmay The bandwidth parameter for the Gaussian kernel on the y-variable set. A positive value. The smaller the smoother.
    numiter Maximum number of iterations.
    reltolstop Convergence threshold. Algorithm stops when relative changes in cost from
                 consecutive iterations is less than the threshold.
Details

Optimization is done by gradient descent, where Nelder-Mead is used for step-size selection. Nelder Mead may fail to increase the cost at times (when stuck at local minima). User may consider restarting the algorithm when this happens.

Value

A list containing:

- \( \mathbf{w}_x \)  
  The canonical projection vector for the x-variable set.
- \( \mathbf{w}_y \)  
  The canonical projection vector for the y-variable set.
- cost  
  A vector of (negative) cost values at each iteration.

Note

Current implementation is slow and requires high storage for large sample data. Sample size > 2000 not recommended.

Author(s)

Billy Chang

References


See Also

hsicCCA

Examples

```r
set.seed(1)
numData <- 100
numDim <- 2
x <- matrix(rnorm(numData*numDim),numData,numDim)
y <- matrix(rnorm(numData*numDim),numData,numDim)
z <- runif(numData,-pi,pi)
y[,1] <- cos(z)+rnorm(numData,sd=0.1); x[,1] <- sin(z)+rnorm(numData,sd=0.1)
x <- scale(x)
y <- scale(y)
fit <- hsicCCAfunc(x,y,sigmax=1,sigmay=1)
plot(x%*%fit$Wx,y%*%fit$Wy)
```
Canonical Correlation Analysis based on the Centered Kernel Target Alignment.

Description

Given two multi-dimensional data sets, find pairs of canonical projection pairs that maximize the Centered Kernel Target Alignment Algorithm.

Usage

\[
\text{ktacca}(x, y, M, \text{sigmax} = \text{NULL}, \text{sigmay} = \text{NULL}, \text{numrepeat} = 5, \text{numiter} = 100, \text{reltolstop} = 1e^{-04})
\]

Arguments

- \(x\) The x-variable data matrix. One row per observation.
- \(y\) The y-variable data matrix. One row per observation.
- \(M\) Number of canonical projection pairs to extract.
- \(\text{sigmax}\) The bandwidth parameter for the Gaussian kernel on the x-variable set. A positive value. The smaller the smoother. If NULL, set to median(dist(x)), and will be updated automatically for extracting different pairs of canonical projection.
- \(\text{sigmay}\) The bandwidth parameter for the Gaussian kernel on the y-variable set. A positive value. The smaller the smoother. If NULL, set to median(dist(y)), and will be updated automatically for extracting different pairs of canonical projection.
- \(\text{numrepeat}\) Number of random restarts.
- \(\text{numiter}\) Maximum number of iterations for extracting each pair of canonical projections.
- \(\text{reltolstop}\) Convergence threshold. Algorithm stops when relative change in cost from consecutive iterations is less than the threshold and will then move on to find the next pair of canonical vectors.

Details

Optimization is done by gradient descent, where Nelder-Mead is used for step-size selection. Nelder Mead may fail to increase the cost at times (when stuck at local minima). User may consider restarting the algorithm when this happens.

Value

A list containing:

- \(\mathbf{W}_x\) The M canonical projection vectors for the x-variable set. Each column corresponds to a projection vector.
- \(\mathbf{W}_y\) The M canonical projection vectors for the y-variable set. Each column corresponds to a projection vector.
**Note**

Current implementation is slow and requires high storage for large sample data. Sample size > 2000 not recommended.

**Author(s)**

Billy Chang

**References**


**See Also**

hsicCCA, ktaCCAfunc

**Examples**

```r
set.seed(1)
numData <- 100
numDim <- 3
x <- matrix(rnorm(numData*numDim),numData,numDim)
y <- matrix(rnorm(numData*numDim),numData,numDim)
z <- runif(numData,-pi,pi)
y[,1] <- cos(z)*rnorm(numData,sd=0.1); x[,1] <- sin(z)*rnorm(numData,sd=0.1)
y[,2] <- x[,2]*rnorm(numData,sd=0.5)
x <- scale(x)
y <- scale(y)

fit <- ktaCCA(x,y,2,numrepeat=2,numiter=10)
par(mfrow=c(1,2))
for (k in 1:2) plot(x%*%fit$Wx[,K],y%*%fit$Wy[,K])
```

---

**ktaccafunc**

*Canonical Correlation Analysis based on the centered kernel target alignment.*

**Description**

Given two multi-dimensional data sets, find a pair of canonical projection pairs that maximizes the kernel alignment criterion. Called by ktaCCA, and intended for internal use, but users may play with it for potential finer controls.

**Usage**

```r
ktaccafunc(x, y, Wx = NULL, Wy = NULL, sigmax, sigmay, numiter = 20, reltolstop = 1e-04)
```
Arguments

- **x**: The x-variable data matrix. One row per observation.
- **y**: The y-variable data matrix. One row per observation.
- **\( \bar{w}_x \)**: Initial projection vector for the x data set. Randomly set if NULL.
- **\( \bar{w}_y \)**: Initial projection vector for the y data set. Randomly set if NULL.
- **\( \text{sigmax} \)**: The bandwidth parameter for the Gaussian kernel on the x-variable set. A positive value. The smaller the smoother.
- **\( \text{sigmay} \)**: The bandwidth parameter for the Gaussian kernel on the y-variable set. A positive value. The smaller the smoother.
- **\( \text{numiter} \)**: Maximum number of iterations.
- **\( \text{reltolstop} \)**: Convergence threshold. Algorithm stops when relative changes in cost from consecutive iterations is less than the threshold.

Details

Optimization is done by gradient descent, where Nelder-Mead is used for step-size selection. Nelder Mead may fail to increase the cost at times (when stuck at local minima). User may consider restarting the algorithm when this happens.

Value

A list containing:

- **\( \bar{w}_x \)**: The canonical projection vector for the x-variable set.
- **\( \bar{w}_y \)**: The canonical projection vector for the y-variable set.
- **cost**: A vector of (negative) cost values at each iteration.

Note

Current implementation is slow and requires high storage for large sample data. Sample size > 2000 not recommended.

Author(s)

Billy Chang

References


See Also

ktCCA
Examples

```r
set.seed(10)
numData <- 100
numDim <- 2
x <- matrix(rnorm(numData*numDim),numData,numDim)
y <- matrix(rnorm(numData*numDim),numData,numDim)
z <- runif(numData,-pi,pi)
y[,1] <- cos(z)+rnorm(numData, sd=0.1); x[,1] <- sin(z)+rnorm(numData, sd=0.1)
x <- scale(x)
y <- scale(y)

fit <- ktaCCAFunc(x,y,sigmax=1,sigmay=1)
plot(x%%fit$Wx,y%%fit$Wy)
```

Description

Given weights matrix \( W_t \), find sum of weighted pairwise outer product of differences, i.e. \( \sum_{i,j} W_{t,ij} (x_i - x_j)(x_i - x_j)^T \). Internal use only.

Usage

```r
sumWtDiff(Wt, x)
```

Arguments

- `Wt` Weight matrix, `nrow(x)`-by-`nrow(x)`
- `x` data matrix, one observation per row.

Value

the weighted sum of outer product of pairwise differences.

Author(s)

Billy Chang
Index

*Topic package
  hsicCCA-package, 2

hsicCCA, 2, 5, 7
hsicCCA-package, 2
hsicCCAfunc, 4, 4

ktaCCA, 4, 6, 8
ktaCCAFunc, 7, 7

sumWtDiff, 9