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 — *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

Package:	hsicCCA
Type:	Package
Version:	1.0
Date:	2013-03-13
License:	GPL-2

Author(s)

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

References

Chang et. al. (2013) Canonical Correlation Analysis based on Hilbert-Schmidt Independence Criterion and Centered Kernel Target Alignment. ICML 2013.

Gretton et. al. (2005) Measuring statistical dependence with Hilbert-Schmidt Norm. In Algorithmic Learning Theory 2005.

Cortes et. al. (2012) Algorithms for learning kernels based on centered alignments. JMLR 13:795-828.

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 canoncial projection vectors for the x-variable set. Each column corresponds to a projection vector.
`Wy`	The M canoncial 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

Chang et. al. (2013) Canonical Correlation Analysis based on Hilbert-Schmidt Independence Criterion and Centered Kernel Target Alignment. ICML 2013.

Gretton et. al. (2005) Measuring statistical dependence with Hilbert-Schmidt Norm. In Algorithmic Learning Theory 2005.

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:

`Wx`	The canoncial projection vector for the x-variable set.
`Wy`	The canoncial 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

Chang et. al. (2013) Canonical Correlation Analysis based on Hilbert-Schmidt Independence Criterion and Centered Kernel Target Alignment. ICML 2013.

Gretton et. al. (2005) Measuring statistical dependence with Hilbert-Schmidt Norm. In Algorithmic Learning Theory 2005.

See Also

hsicCCA

Examples

```
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)
```

ktaCCA *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

```
ktaCCA(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 canoncial projection vectors for the x-variable set. Each column corresponds to a projection vector.
`Wy`	The M canoncial 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

Chang et. al. (2013) Canonical Correlation Analysis based on Hilbert-Schmidt Independence Criterion and Centered Kernel Target Alignment. ICML 2013.

Cortes et. al. (2012) Algorithms for learning kernels based on centered alignments. JMLR 13:795-828.

See Also

hsicCCA, ktaCCAfunc

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 <- 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

```
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.
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:

Wx	The canoncial projection vector for the x-variable set.
Wy	The canoncial 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

Chang et. al. (2013) Canonical Correlation Analysis based on Hilbert-Schmidt Independence Criterion and Centered Kernel Target Alignment. ICML 2013.

Cortes et. al. (2012) Algorithms for learning kernels based on centered alignments. JMLR 13:795-828.

See Also

ktaCCA

Examples

```
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)
```

`sumWtDiff` *Sum of Weighted Pairwise Outer Differences.*

Description

Given weights matrix Wt, find sum of weighted pairwise outer product of differences, i.e. sum_i,j Wt_ij(x_i-x_j)(x_i-x_j)^T. Internal use only.

Usage

```
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