Package ‘EDMeasure’
February 25, 2018

Title Energy-Based Dependence Measures
Version 1.2.0
Date 2018-01-30

Description Implementations of (1) mutual dependence measures and mutual independence tests in
(2) independent component analysis methods based on mutual dependence measures in
(3) conditional mean dependence measures and conditional mean independence tests in
Shao, X., and Zhang, J. (2014) <doi:10.1080/01621459.2014.887012>,

Depends R (>= 3.4.0)
Imports energy (>= 1.7-0), dHSIC (>= 2.0), rBayesianOptimization (>= 1.1.0)
Suggests testthat (>= 2.0.0)
License GPL (>= 2)
LazyData true
RoxygenNote 6.0.1

NeedsCompilation yes
Author Ze Jin [aut, cre],
Shun Yao [aut],
David S. Matteson [aut],
Xiaofeng Shao [aut]
Maintainer Ze Jin <zj58@cornell.edu>
Repository CRAN
Date/Publication 2018-02-25 22:49:47 UTC
Description

EDMeasure: A package for energy-based dependence measures

Details

The EDMeasure package provides measures of mutual dependence and tests of mutual independence, independent component analysis methods based on mutual dependence measures, and measures of conditional mean dependence and tests of conditional mean independence.

The three main parts are:

- mutual dependence measures via energy statistics
  - measuring mutual dependence
  - testing mutual independence
- independent component analysis via mutual dependence measures
  - applying mutual dependence measures
  - initializing local optimization methods
- conditional mean dependence measures via energy statistics
  - measuring conditional mean dependence
  - testing conditional mean independence

Mutual Dependence Measures via Energy Statistics

Measuring mutual dependence

The mutual dependence measures include:

- asymmetric measure $R_n$ based on distance covariance $V_n$
- symmetric measure $S_n$ based on distance covariance $V_n$
• complete measure $Q_n$ based on complete V-statistics
• simplified complete measure $Q^*_n$ based on incomplete V-statistics
• asymmetric measure $J_n$ based on complete measure $Q_n$
• simplified asymmetric measure $J^*_n$ based on simplified complete measure $Q^*_n$
• symmetric measure $I_n$ based on complete measure $Q_n$
• simplified symmetric measure $I^*_n$ based on simplified complete measure $Q^*_n$

**Testing mutual independence**
The mutual independence tests based on the mutual dependence measures are implemented as permutation tests.

**Independent Component Analysis via Mutual Dependence Measures**

**Applying mutual dependence measures**
The mutual dependence measures include:
• distance-based energy statistics
  – asymmetric measure $R_n$ based on distance covariance $V_n$
  – symmetric measure $S_n$ based on distance covariance $V_n$
  – simplified complete measure $Q^*_n$ based on incomplete V-statistics
• kernel-based maximum mean discrepancies
  – d-variable Hilbert–Schmidt independence criterion $dHSIC_n$ based on Hilbert–Schmidt independence criterion $HSIC_n$

**Initializing local optimization methods**
The initialization methods include:
• Latin hypercube sampling
• Bayesian optimization

**Conditional Mean Dependence Measures via Energy Statistics**

**Measuring conditional mean dependence**
The conditional mean dependence measures include:
• conditional mean dependence of $Y$ given $X$
  – martingale difference divergence
  – martingale difference correlation
  – martingale difference divergence matrix
• conditional mean dependence of $Y$ given $X$ adjusting for the dependence on $Z$
  – partial martingale difference divergence
  – partial martingale difference correlation
Testing conditional mean independence

The conditional mean independence tests include:

- conditional mean independence of $Y$ given $X$ conditioning on $Z$
  - martingale difference divergence under a linear assumption
  - partial martingale difference divergence

The conditional mean independence tests based on the conditional mean dependence measures are implemented as permutation tests.

Author(s)

Ze Jin <zj58@cornell.edu>, Shun Yao <shunyao2@illinois.edu>,
David S. Matteson <matteson@cornell.edu>, Xiaofeng Shao <xshao@illinois.edu>

---

### cmdm_test

**Conditional Mean Independence Tests**

<table>
<thead>
<tr>
<th>Description</th>
<th>cmdm_test tests conditional mean independence of $Y$ given $X$ conditioning on $Z$, where each contains one variable (univariate) or more variables (multivariate). All tests are implemented as permutation tests.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Usage</strong></td>
<td>cmdm_test(X, Y, Z, num_perm = 500, type = &quot;linmdd&quot;, compute = &quot;C&quot;, center = &quot;U&quot;)</td>
</tr>
<tr>
<td><strong>Arguments</strong></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>A vector, matrix or data frame, where rows represent samples, and columns represent variables.</td>
</tr>
<tr>
<td>Y</td>
<td>A vector, matrix or data frame, where rows represent samples, and columns represent variables.</td>
</tr>
<tr>
<td>Z</td>
<td>A vector, matrix or data frame, where rows represent samples, and columns represent variables.</td>
</tr>
<tr>
<td>num_perm</td>
<td>The number of permutation samples drawn to approximate the asymptotic distributions of mutual dependence measures.</td>
</tr>
<tr>
<td>type</td>
<td>The type of conditional mean dependence measures, including</td>
</tr>
<tr>
<td></td>
<td>• linmdd: martingale difference divergence under a linear assumption;</td>
</tr>
<tr>
<td></td>
<td>• pmdd: partial martingale difference divergence.</td>
</tr>
<tr>
<td>compute</td>
<td>The computation method for martingale difference divergence, including</td>
</tr>
<tr>
<td></td>
<td>• C: computation implemented in C code;</td>
</tr>
<tr>
<td></td>
<td>• R: computation implemented in R code.</td>
</tr>
</tbody>
</table>
The centering approach for martingale difference divergence, including
- U: U-centering which leads to an unbiased estimator;
- D: double-centering which leads to a biased estimator.

Value
cmdm_test returns a list including the following components:
stat The value of the conditional mean dependence measure.
dist The p-value of the conditional mean independence test.

References

Examples
```r
## Not run:
# X, Y, Z are vectors with 10 samples and 1 variable
X <- rnorm(10)
Y <- rnorm(10)
Z <- rnorm(10)
cmdm_test(X, Y, Z, type = "linmdd")

# X, Y, Z are 10 x 2 matrices with 10 samples and 2 variables
X <- matrix(rnorm(10 * 2), 10, 2)
Y <- matrix(rnorm(10 * 2), 10, 2)
Z <- matrix(rnorm(10 * 2), 10, 2)
cmdm_test(X, Y, Z, type = "pmdd")
## End(Not run)
```

mdc Martingale Difference Correlation

Description
mdc measures conditional mean dependence of Y given X, where each contains one variable (univariate) or more variables (multivariate).

Usage
mdc(X, Y, center = "U")
Arguments

X A vector, matrix or data frame, where rows represent samples, and columns represent variables.

Y A vector, matrix or data frame, where rows represent samples, and columns represent variables.

center The approach for centering, including
  • U: U-centering which leads to an unbiased estimator;
  • D: double-centering which leads to a biased estimator.

Value

mdc returns the squared martingale difference correlation of Y given X.

References


Examples

# X, Y are 10 x 2 matrices with 10 samples and 2 variables
X <- matrix(rnorm(10 * 2), 10, 2)
Y <- matrix(rnorm(10 * 2), 10, 2)

mdc(X, Y, center = "U")
mdc(X, Y, center = "D")

---

mdd Martingale Difference Divergence

Description

mdd measures conditional mean dependence of Y given X, where each contains one variable (univariate) or more variables (multivariate).

Usage

mdd(X, Y, compute = "C", center = "U")
Arguments

- **X**: A vector, matrix or data frame, where rows represent samples, and columns represent variables.
- **Y**: A vector, matrix or data frame, where rows represent samples, and columns represent variables.
- **compute**: The method for computation, including
  - **C**: computation implemented in C code;
  - **R**: computation implemented in R code.
- **center**: The approach for centering, including
  - **U**: U-centering which leads to an unbiased estimator;
  - **D**: double-centering which leads to a biased estimator.

Value

`mdd` returns the squared martingale difference divergence of `Y` given `X`.

References


Examples

```r
# X, Y are vectors with 10 samples and 1 variable
X <- rnorm(10)
Y <- rnorm(10)

mdd(X, Y, compute = "C")
mdd(X, Y, compute = "R")

# X, Y are 10 x 2 matrices with 10 samples and 2 variables
X <- matrix(rnorm(10 * 2), 10, 2)
Y <- matrix(rnorm(10 * 2), 10, 2)

mdd(X, Y, center = "U")
mdd(X, Y, center = "D")
```
martingale difference divergence from a scalar to a matrix. It encodes the linear combinations of all univariate components in \( Y \) that are conditionally mean independent of \( X \). Only the double-centering approach is applied.

Usage

\[
\text{mddm}(X, Y, \text{compute} = "C")
\]

Arguments

- **X**: A vector, matrix or data frame, where rows represent samples, and columns represent variables.
- **Y**: A vector, matrix or data frame, where rows represent samples, and columns represent variables.
- **compute**: The method for computation, including
  - **C**: computation implemented in C code;
  - **R**: computation implemented in R code.

Value

mddm returns the martingale difference divergence matrix of \( Y \) given \( X \).

References


Examples

```r
# X, Y are vectors with 10 samples and 1 variable
X <- rnorm(10)
Y <- rnorm(10)

mddm(X, Y, compute = "C")
mddm(X, Y, compute = "R")

# X, Y are 10 x 2 matrices with 10 samples and 2 variables
X <- matrix(rnorm(10 * 2), 10, 2)
Y <- matrix(rnorm(10 * 2), 10, 2)

mddm(X, Y, compute = "C")
mddm(X, Y, compute = "R")
```
**mdm**

*Mutual Dependence Measures*

**Description**

`mdm` measures mutual dependence of all components in `X`, where each component contains one variable (univariate) or more variables (multivariate).

**Usage**

```r
mdm(X, dim_comp = NULL, dist_comp = FALSE, type = "comp_simp")
```

**Arguments**

- `X` A matrix or data frame, where rows represent samples, and columns represent variables.
- `dim_comp` The numbers of variables contained by all components in `X`. If omitted, each component is assumed to contain exactly one variable.
- `dist_comp` Logical. If `TRUE`, the distances between all components from all samples in `X` will be returned.
- `type` The type of mutual dependence measures, including:
  - `asym_dcov`: asymmetric measure $R_n$ based on distance covariance $V_n$;
  - `sym_dcov`: symmetric measure $S_n$ based on distance covariance $V_n$;
  - `comp`: complete measure $Q_n$ based on complete V-statistics;
  - `comp_simp`: simplified complete measure $Q^*_n$ based on incomplete V-statistics;
  - `asym_comp`: asymmetric measure $J_n$ based on complete measure $Q_n$;
  - `asym_comp_simp`: simplified asymmetric measure $J^*_n$ based on simplified complete measure $Q^*_n$;
  - `sym_comp`: symmetric measure $I_n$ based on complete measure $Q_n$;
  - `sym_comp_simp`: simplified symmetric measure $I^*_n$ based on simplified complete measure $Q^*_n$.

From experiments, `asym_dcov`, `sym_dcov`, `comp_simp` are recommended.

**Value**

`mdm` returns a list including the following components:

- `stat` The value of the mutual dependence measure.
- `dist` The distances between all components from all samples.

**References**

Examples

```r
# X is a 10 x 3 matrix with 10 samples and 3 variables
X <- matrix(rnorm(10 * 3), 10, 3)

# assume X = (X1, X2) where X1 is 1-dim, X2 is 2-dim
mdm(X, dim_comp = c(1, 2), type = "asym_dcov")

# assume X = (X1, X2) where X1 is 2-dim, X2 is 1-dim
mdm(X, dim_comp = c(2, 1), type = "sym_dcov")

# assume X = (X1, X2, X3) where X1 is 1-dim, X2 is 1-dim, X3 is 1-dim
mdm(X, dim_comp = c(1, 1, 1), type = "comp_simp")
```

mdm_ica

Independent Component Analysis via Mutual Dependence Measures

Description

mdm_ica performs independent component analysis by minimizing mutual dependence measures of all univariate components in X.

Usage

```r
mdm_ica(X, num_lhs = NULL, type = "comp", num_bo = NULL, kernel = "exp", algo = "par")
```

Arguments

- **X**: A matrix or data frame, where rows represent samples, and columns represent components.
- **num_lhs**: The number of points generated by Latin hypercube sampling. If omitted, an adaptive number is used.
- **type**: The type of mutual dependence measures, including
  - `asym`: asymmetric measure $R_n$ based on distance covariance $V_n$;
  - `sym`: symmetric measure $S_n$ based on distance covariance $V_n$;
  - `comp`: simplified complete measure $Q_n^*$ based on incomplete V-statistics;
  - `dhsic`: d-variable Hilbert–Schmidt independence criterion $dHSIC_n$ based on Hilbert–Schmidt independence criterion $HSIC_n$.
- **num_bo**: The number of points evaluated by Bayesian optimization.
- **kernel**: The kernel of the underlying Gaussian process in Bayesian optimization, including
  - `exp`: squared exponential kernel;
  - `mat`: Matern 5/2 kernel.
- **algo**: The algorithm of optimization, including
  - `def`: deflation algorithm, where the components are extracted one at a time;
  - `par`: parallel algorithm, where the components are extracted simultaneously.
Value

`mdm_ica` returns a list including the following components:

- **theta**: The rotation angles of the estimated unmixing matrix.
- **W**: The estimated unmixing matrix.
- **obj**: The objective value of the estimated independence components.
- **S**: The estimated independence components.

References


Examples

```r
# X is a 10 x 3 matrix with 10 samples and 3 components
X <- matrix(rnorm(10 * 3), 10, 3)

# deflation algorithm
mdm_ica(X, type = "asym", algo = "def")
# parallel algorithm
mdm_ica(X, type = "asym", algo = "par")

## Not run:
# bayesian optimization with exponential kernel
mdm_ica(X, type = "sym", num_bo = 1, kernel = "exp", algo = "par")
# bayesian optimization with matern kernel
mdm_ica(X, type = "comp", num_bo = 1, kernel = "mat", algo = "par")

## End(Not run)
```

---

### mdm_test  
**Mutual Independence Tests**

**Description**

`mdm_test` tests mutual independence of all components in `X`, where each component contains one variable (univariate) or more variables (multivariate). All tests are implemented as permutation tests.

**Usage**

```r
mdm_test(X, dim_comp = NULL, num_perm = NULL, type = "comp_simp")
```
Arguments

X
A matrix or data frame, where rows represent samples, and columns represent variables.

dim_comp
The numbers of variables contained by all components in X. If omitted, each component is assumed to contain exactly one variable.

num_perm
The number of permutation samples drawn to approximate the asymptotic distributions of mutual dependence measures. If omitted, an adaptive number is used.

type
The type of mutual dependence measures, including

- \text{asym\_dcov}: asymmetric measure \( R_n \) based on distance covariance \( V_n \);
- \text{sym\_dcov}: symmetric measure \( S_n \) based on distance covariance \( V_n \);
- \text{comp}: complete measure \( Q_n \) based on complete V-statistics;
- \text{comp\_simp}: simplified complete measure \( Q_n^{*} \) based on incomplete V-statistics;
- \text{asym\_comp}: asymmetric measure \( J_n \) based on complete measure \( Q_n \);
- \text{asym\_comp\_simp}: simplified asymmetric measure \( J_n^{*} \) based on simplified complete measure \( Q_n^{*} \);
- \text{sym\_comp}: symmetric measure \( I_n \) based on complete measure \( Q_n \);
- \text{sym\_comp\_simp}: simplified symmetric measure \( I_n^{*} \) based on simplified complete measure \( Q_n^{*} \).

From experiments, \text{asym\_dcov}, \text{sym\_dcov}, \text{comp\_simp} are recommended.

Value

\text{mdm\_test} returns a list including the following components:

- \text{stat}
  The value of the mutual dependence measure.

- \text{pval}
  The p-value of the mutual independence test.

References


Examples

```r
## Not run:
# X is a 10 \times 3 matrix with 10 samples and 3 variables
X <- matrix(rnorm(10 * 3), 10, 3)

# assume X = (X1, X2) where X1 is 1-dim, X2 is 2-dim
mdm_test(X, dim_comp = c(1, 2), type = "asym\_dcov")

# assume X = (X1, X2) where X1 is 2-dim, X2 is 1-dim
mdm_test(X, dim_comp = c(2, 1), type = "sym\_dcov")

# assume X = (X1, X2, X3) where X1 is 1-dim, X2 is 1-dim, X3 is 1-dim
```
pmdc

```r
mdm_test(X, dim_comp = c(1, 1, 1), type = "comp_simp")
## End(Not run)
```

---

pmdc Partial Martingale Difference Correlation

**Description**

pmdc measures conditional mean dependence of $Y$ given $X$ adjusting for the dependence on $Z$, where each contains one variable (univariate) or more variables (multivariate). Only the U-centering approach is applied.

**Usage**

```r
pmdc(X, Y, Z)
```

**Arguments**

- **X** A vector, matrix or data frame, where rows represent samples, and columns represent variables.
- **Y** A vector, matrix or data frame, where rows represent samples, and columns represent variables.
- **Z** A vector, matrix or data frame, where rows represent samples, and columns represent variables.

**Value**

pmdc returns the squared partial martingale difference correlation of $Y$ given $X$ adjusting for the dependence on $Z$.

**References**


**Examples**

```r
# X, Y, Z are 10 x 2 matrices with 10 samples and 2 variables
X <- matrix(rnorm(10 * 2), 10, 2)
Y <- matrix(rnorm(10 * 2), 10, 2)
Z <- matrix(rnorm(10 * 2), 10, 2)

pmdc(X, Y, Z)
```
Partial Martingale Difference Divergence

Description

`pmdd` measures conditional mean dependence of \( Y \) given \( X \) adjusting for the dependence on \( Z \), where each contains one variable (univariate) or more variables (multivariate). Only the U-centering approach is applied.

Usage

```r
pmdd(X, Y, Z)
```

Arguments

- **X**: A vector, matrix or data frame, where rows represent samples, and columns represent variables.
- **Y**: A vector, matrix or data frame, where rows represent samples, and columns represent variables.
- **Z**: A vector, matrix or data frame, where rows represent samples, and columns represent variables.

Value

`pmdd` returns the squared partial martingale difference divergence of \( Y \) given \( X \) adjusting for the dependence on \( Z \).

References


Examples

```r
# X, Y, Z are vectors with 10 samples and 1 variable
X <- rnorm(10)
Y <- rnorm(10)
Z <- rnorm(10)

pmdd(X, Y, Z)

# X, Y, Z are 10 x 2 matrices with 10 samples and 2 variables
X <- matrix(rnorm(10 * 2), 10, 2)
Y <- matrix(rnorm(10 * 2), 10, 2)
Z <- matrix(rnorm(10 * 2), 10, 2)

pmdd(X, Y, Z)
```
Index

cmdm_test, 4

EDMeasure (EDMeasure-package), 2
EDMeasure-package, 2

mdc, 5
mdd, 6
mddm, 8
mdm, 9
mdm_ica, 10
mdm_test, 11

pmdc, 13
pmdd, 14