Package 'MTSYS'

January 20, 2025

Type Package

Title Methods in Mahalanobis-Taguchi (MT) System

Version 1.2.0

Date 2017-07-28

Description Mahalanobis-Taguchi (MT) system is a collection of multivariate analysis methods developed for the field of quality engineering. MT system consists of two families depending on their purpose. One is a family of Mahalanobis-Taguchi (MT) methods (in the broad sense) for diagnosis (see Woodall, W. H., Koudelik, R., Tsui, K. L., Kim, S. B., Stoumbos, Z. G., and Carvounis, C. P. (2003) <doi:10.1198/004017002188618626>) and the other is a family of Taguchi (T) methods for forecasting (see Kawada, H., and Nagata, Y. (2015) <doi:10.17929/tqs.1.12>). The MT package contains three basic methods for the family of MT methods and one basic method for the family of T methods. The MT method (in the narrow sense), the Mahalanobis-Taguchi Adjoint (MTA) methods, and the Recognition-Taguchi (RT) method are for the MT method and the two-sided Taguchi (T1) method is for the family of T methods. In addition, the Ta and Tb methods, which are the improved versions of the T1 method, are included.

Depends R (>= 2.10)

Imports stats

Suggests testthat, covr

Encoding UTF-8

License MIT + file LICENSE

RoxygenNote 5.0.1

LazyData true

URL https://github.com/okayaa/MTSYS

BugReports https://github.com/okayaa/MTSYS/issues

Author Akifumi Okayama [aut, cre],
Masato Ohkubo [ctb],
Yasushi Nagata [ctb]

Maintainer Akifumi Okayama <akifumi.okayama@akane.waseda.jp>

NeedsCompilation no

Repository CRAN

Date/Publication 2017-09-10 16:44:06 UTC

Contents

calc_cofactor 2
calc_M_hat 3
calc_overall_predicton_eta 4
diagnosis 5
diagnosis.MT 6
diagnosis.MTA 8
diagnosis.RT 10
forecasting 11
forecasting.T1 12
forecasting.Ta 14
forecasting.Tb 15
general_diagnosis.MT 16
general_forecasting.T 18
general_MT 19
general_T 21
generates_dimensionality_reduction_function 23
generates_model 24
generates_normalization_function 26
generates_transformation_functions_T1 27
generates_transformation_functions_Tb 28
generates_unit_space 29
MT 31
MTA 32
RT 34
T1 35
Ta 37
Tb 39

Index **41**

`calc_cofactor` *Function to calculate a cofactor matrix*

Description

`calc_cofactor` calculates a cofactor matrix.

Usage

```
calc_cofactor(data)
```

Arguments

`data` Matrix with n rows (samples) and p columns (variables). All data should be continuous values and should not have missing values.

Value

`calc_cofactor` returns a cofactor matrix of size p x p.

See Also

MTA

Examples

```
# 40 data for versicolor in the iris dataset
iris_versicolor <- iris[61:100, -5]

calc_cofactor(cov(iris_versicolor))
```

`calc_M_hat` *Function to estimate M value (M hat) for a family of T methods.*

Description

`calc_M_hat` estimates M values (M hat) for the T method.

Usage

```
calc_M_hat(X, beta_hat, eta_hat)
```

Arguments

`X` Matrix with n rows (samples) and q columns (variables). The independent variable data after the data transformation. All data should be continuous values and should not have missing values.

`beta_hat` Vector with length q. Estimated proportionality constants between each independent variable and the dependent variable.

`eta_hat` Vector with length q. Estimated squared signal-to-noise ratios (S/N) corresponding to `beta_hat`.

Value

Vector with length n. Estimated M values (M hat).

See Also

`general_T` and `general_forecasting.T`

Examples

```
# The value of the dependent variable of the following samples mediates
# in the stackloss dataset.
stackloss_center <- stackloss[c(9, 10, 11, 20, 21), ]

# The following samples are data other than the unit space data and the test
# data.
stackloss_signal <- stackloss[-c(2, 9, 10, 11, 12, 19, 20, 21), ]

# The following settings are same as the T1 method.
model <- general_T(unit_space_data = stackloss_center,
                   signal_space_data = stackloss_signal,
                   generates_transform_functions =
                                       generates_transformation_functions_T1,
                   includes_transformed_data = TRUE)

modified_eta_hat <- model$eta_hat
modified_eta_hat[3] <- 0

(modified_M_hat <- calc_M_hat(model$X, model$beta_hat, modified_eta_hat))
```

---

calc_overall_predicton_eta

Function to calculate overall prediction eta for the T method

---

Description

`calc_M_hat` calculates the overall prediction eta for the T method.

Usage

```
calc_overall_predicton_eta(M, M_hat, subtracts_V_e = TRUE)
```

Arguments

`M`	Vector with length n. The (true) value of the dependent variable after the data trasformation.
`M_hat`	Vector with length n. The estimated values of the dependent variable after the data trasformation.
`subtracts_V_e`	If `TRUE`, then the error variance is subtracted in the numerator when calculating `eta_hat`.

Value

Numeric. Overall prediction eta which is used to measure the estimation accuracy.

See Also

general_T and general_forecasting.T

Examples

```
# The value of the dependent variable of the following samples mediates
# in the stackloss dataset.
stackloss_center <- stackloss[c(9, 10, 11, 20, 21), ]

# The following samples are data other than the unit space data and the test
# data.
stackloss_signal <- stackloss[-c(2, 9, 10, 11, 12, 19, 20, 21), ]

# The following settings are same as the T1 method.
model <- general_T(unit_space_data = stackloss_center,
                   signal_space_data = stackloss_signal,
                   generates_transform_functions =
                                       generates_transformation_functions_T1,
                   subtracts_V_e = TRUE,
                   includes_transformed_data = TRUE)

modified_eta_hat <- model$eta_hat
modified_eta_hat[3] <- 0

modified_M_hat <- calc_M_hat(model$X, model$beta_hat, modified_eta_hat)

(modified_overall_predicton_eta <-
                            calc_overall_predicton_eta(model$M,
                                                       modified_M_hat,
                                                       subtracts_V_e = TRUE))
```

---

diagnosis *Function to predict a diagnosis for a family of Mahalanobis-Taguchi (MT) methods*

Description

diagnosis is a generic function. For details, see diagnosis.MT, diagnosis.MTA, diagnosis.RT or general_diagnosis.MT.

Usage

```
diagnosis(unit_space, newdata, threshold, includes_transformed_newdata)
```

Arguments

unit_space	Object generated as a unit space.
newdata	Matrix with n rows (samples) and p columns (variables). The data are used to calculate the desired distances from the unit space. All data should be continuous values and should not have missing values.
threshold	Numeric specifying the threshold value to classify each sample into positive (TRUE) or negative (FALSE).
includes_transformed_newdata	If TRUE, then the transformed data for newdata are included in a return object.

Value

A list containing the following components is returned.

distance	Vector with length n. Distances from the unit space to each sample.
le_threshold	Vector with length n. Logical values indicating the distance of each sample is less than or equal to the threhold value (TRUE) or not (FALSE).
threshold	Numeric value to classify the sample into positive or negative.
unit_space	Object passed by unit_space.
n	The number of samples for newdata.
q	The number of variables after the data transformation.
x	If includes_transformed_newdata is TRUE, then the transformed data for newdata are included.

See Also

diagnosis.MT, diagnosis.MTA, and diagnosis.RT

---

diagnosis.MT — *Diagnosis method for the Mahalanobis-Taguchi (MT) method*

Description

diagnosis.MT (via diagnosis) calculates the mahalanobis distance based on the unit space generated by MT or generates_unit_space(..., method = "MT") and classifies each sample into positive (TRUE) or negative (FALSE) by comparing the values with the set threshold value.

Usage

```
## S3 method for class 'MT'
diagnosis(unit_space, newdata, threshold = 4,
  includes_transformed_newdata = FALSE)
```

Arguments

unit_space	Object of class "MT" generated by MT or generates_unit_space(..., method = "MT").
newdata	Matrix with n rows (samples) and p columns (variables). The data are used to calculate the desired distances from the unit space. All data should be continuous values and should not have missing values.
threshold	Numeric specifying the threshold value to classify each sample into positive (TRUE) or negative (FALSE).
includes_transformed_newdata	If TRUE, then the transformed data for newdata are included in a return object.

Value

diagnosis.MT (via diagnosis) returns a list containing the following components:

distance	Vector with length n. Distances from the unit space to each sample.
le_threshold	Vector with length n. Logical values indicating the distance of each sample is less than or equal to the threhold value (TRUE) or not (FALSE).
threshold	Numeric value to classify the sample into positive or negative.
unit_space	Object of class "MT" passed by unit_space.
n	The number of samples for newdata.
q	The number of variables after the data transformation. q equals p.
x	If includes_transformed_newdata is TRUE, then the transformed data for newdata are included.

References

Taguchi, G. (1995). Pattern Recognition and Quality Engineering (1). *Journal of Quality Engineering Society, 3*(2), 2-5. (In Japanese)

Taguchi, G., Wu, Y., & Chodhury, S. (2000). *Mahalanobis-Taguchi System.* McGraw-Hill Professional.

Taguchi, G., & Jugulum, R. (2002). *The Mahalanobis-Taguchi strategy: A pattern technology system.* John Wiley & Sons.

Woodall, W. H., Koudelik, R., Tsui, K. L., Kim, S. B., Stoumbos, Z. G., & Carvounis, C. P. (2003). A review and analysis of the Mahalanobis-Taguchi system. *Technometrics, 45*(1), 1-15.

See Also

general_diagnosis.MT and MT

Examples

```
# 40 data for versicolor in the iris dataset
iris_versicolor <- iris[61:100, -5]

unit_space_MT <- MT(unit_space_data = iris_versicolor,
                    includes_transformed_data = TRUE)

# 10 data for each kind (setosa, versicolor, virginica) in the iris dataset
iris_test <- iris[c(1:10, 51:60, 101:111), -5]

diagnosis_MT <- diagnosis(unit_space = unit_space_MT,
                          newdata = iris_test,
                          threshold = 4,
                          includes_transformed_newdata = TRUE)

(diagnosis_MT$distance)
(diagnosis_MT$le_threshold)
```

---

diagnosis.MTA *Diagnosis method for the Mahalanobis-Taguchi Adjoint (MTA) method*

Description

`diagnosis.MTA` (via `diagnosis`) calculates the distance based on the unit space generated by `MTA` or `generates_unit_space`(..., method = "MTA") and classifies each sample into positive (`TRUE`) or negative (`FALSE`) by comparing the values with the set threshold value.

Usage

```
## S3 method for class 'MTA'
diagnosis(unit_space, newdata, threshold,
  includes_transformed_newdata = FALSE)
```

Arguments

`unit_space`	Object of class "MTA" generated by `MTA` or `generates_unit_space`(..., method = "MTA").
`newdata`	Matrix with n rows (samples) and p columns (variables). The data are used to calculate the desired distances from the unit space. All data should be continuous values and should not have missing values.
`threshold`	Numeric specifying the threshold value to classify each sample into positive (`TRUE`) or negative (`FALSE`).
`includes_transformed_newdata`	If `TRUE`, then the transformed data for `newdata` are included in a return object.

Value

diagnosis.MTA (via diagnosis) returns a list containing the following components:

distance	Vector with length n. Distances from the unit space to each sample.
le_threshold	Vector with length n. Logical values indicating the distance of each sample is less than or equal to the threhold value (TRUE) or not (FALSE).
threshold	Numeric value to classify the sample into positive or negative.
unit_space	Object of class "MTA" passed by unit_space.
n	The number of samples for newdata.
q	The number of variables after the data transformation. q equals p.
x	If includes_transformed_newdata is TRUE, then the transformed data for newdata are included.

References

Taguchi, G. & Kanetaka, T. (2002). *Engineering Technical Development in MT System - Lecture on Applied Quality.* Japanese Standards Association. (In Japanese)

Taguchi, G., & Jugulum, R. (2002). *The Mahalanobis-Taguchi strategy: A pattern technology system.* John Wiley & Sons.

See Also

general_diagnosis.MT and MTA

Examples

```
# 40 data for versicolor in the iris dataset
iris_versicolor <- iris[61:100, -5]

unit_space_MTA <- MTA(unit_space_data = iris_versicolor,
                      includes_transformed_data = TRUE)

# 10 data for each kind (setosa, versicolor, virginica) in the iris dataset
iris_test <- iris[c(1:10, 51:60, 101:111), -5]

diagnosis_MTA <- diagnosis(unit_space = unit_space_MTA,
                           newdata = iris_test,
                           threshold = 0.5,
                           includes_transformed_newdata = TRUE)

(diagnosis_MTA$distance)
(diagnosis_MTA$le_threshold)
```

diagnosis.RT *Diagnosis method for the Recognition-Taguchi (RT) method*

Description

diagnosis.RT (via diagnosis) calculates the distance based on the unit space generated by RT or generates_unit_space(..., method = "RT") and classifies each sample into positive (TRUE) or negative (FALSE) by comparing the values with the set threshold value.

Usage

```
## S3 method for class 'RT'
diagnosis(unit_space, newdata, threshold,
  includes_transformed_newdata = FALSE)
```

Arguments

unit_space
: Object of class "RT" generated by RT or generates_unit_space(..., method = "RT").

newdata
: Matrix with n rows (samples) and p columns (variables). The data are used to calculate the desired distances from the unit space. All data should be continuous values and should not have missing values.

threshold
: Numeric specifying the threshold value to classify each sample into positive (TRUE) or negative (FALSE).

includes_transformed_newdata
: If TRUE, then the transformed data for newdata are included in a return object.

Value

diagnosis.RT (via diagnosis) returns a list containing the following components:

distance
: Vector with length n. Distances from the unit space to each sample.

le_threshold
: Vector with length n. Logical values indicating the distance of each sample is less than or equal to the threhold value (TRUE) or not (FALSE).

threshold
: Numeric value to classify the sample into positive or negative.

unit_space
: Object of class "RT" passed by unit_space.

n
: The number of samples for newdata.

q
: The number of variables after the data transformation. q is always 2.

x
: If includes_transformed_newdata is TRUE, then the transformed data for newdata are included.

References

Taguchi, G. (2006). Objective Function and Generic Function (11). *Journal of Quality Engineering Society, 14*(2), 5-9. (In Japanese)

Huda, F., Kajiwara, I., Hosoya, N., & Kawamura, S. (2013). Bolt loosening analysis and diagnosis by non-contact laser excitation vibration tests. *Mechanical systems and signal processing, 40*(2), 589-604.

See Also

general_diagnosis.MT and RT

Examples

```
# 40 data for versicolor in the iris dataset
iris_versicolor <- iris[61:100, -5]

unit_space_RT <- RT(unit_space_data = iris_versicolor,
                    includes_transformed_data = TRUE)

# 10 data for each kind (setosa, versicolor, virginica) in the iris dataset
iris_test <- iris[c(1:10, 51:60, 101:111), -5]

diagnosis_RT <- diagnosis(unit_space = unit_space_RT,
                          newdata = iris_test,
                          threshold = 0.2,
                          includes_transformed_newdata = TRUE)

(diagnosis_RT$distance)
(diagnosis_RT$le_threshold)
```

`forecasting` *Function to predict a forecasting for a family of Taguchi (T) methods*

Description

`forecasting` is a generic function. For details, see `forecasting.T1`, `forecasting.Ta`, `forecasting.Tb` or `general_forecasting.T`.

Usage

```
forecasting(model, newdata, includes_transformed_newdata)
```

Arguments

`model` Object generated as a model.

`newdata` Matrix with n rows (samples) and p columns (variables). The Data to be estimated. All data should be continuous values and should not have missing values.

`includes_transformed_newdata` If `TRUE`, then the transformed data for `newdata` are included in a return object.

Value

A list containing the following components is returned.

`M_hat` Vector with length n. The estimated values of the dependent variable after the data trasformation.

`y_hat` Vector with length n. The estimated values after the inverse transformation from `M_hat`.

`model` Object passed by `model`.

`n` The number of samples for `newdata`.

`q` The number of variables after the data transformation.

`X` If `includes_transformed_newdata` is `TRUE`, then the transformed data for `newdata` are included.

See Also

forecasting.T1, forecasting.Ta, and forecasting.Tb

---

forecasting.T1 *Forecasting method for the T1 method*

Description

`forecasting.T1` (via forecasting) estimates the dependent values based on the T1 model.

Usage

```
## S3 method for class 'T1'
forecasting(model, newdata, includes_transformed_newdata = FALSE)
```

Arguments

`model` Object of class "T1" generated by T1 or generates_model(..., method = "T1").

`newdata` Matrix with n rows (samples) and p columns (variables). The Data to be estimated. All data should be continuous values and should not have missing values.

`includes_transformed_newdata` If `TRUE`, then the transformed data for `newdata` are included in a return object.

Value

A list containing the following components is returned.

M_hat	Vector with length n. The estimated values of the dependent variable after the data transformation.
y_hat	Vector with length n. The estimated values after the inverse transformation from M_hat.
model	Object of class "T1" passed by model.
n	The number of samples for newdata.
q	The number of variables after the data transformation. q equals p.
X	If includes_transformed_newdata is TRUE, then the transformed data for newdata are included.

References

Taguchi, G. (2006). Objective Function and Generic Function (12). *Journal of Quality Engineering Society, 14*(3), 5-9. (In Japanese)

Inou, A., Nagata, Y., Horita, K., & Mori, A. (2012). Prediciton Accuracies of Improved Taguchi's T Methods Compared to those of Multiple Regresssion Analysis. *Journal of the Japanese Society for Quality Control, 42*(2), 103-115. (In Japanese)

Kawada, H., & Nagata, Y. (2015). An application of a generalized inverse regression estimator to Taguchi's T-Method. *Total Quality Science, 1*(1), 12-21.

See Also

general_forecasting.T and T1

Examples

```
# The value of the dependent variable of the following samples mediates
# in the stackloss dataset.
stackloss_center <- stackloss[c(9, 10, 11, 20, 21), ]

# The following samples are data other than the unit space data and the test
# data.
stackloss_signal <- stackloss[-c(2, 9, 10, 11, 12, 19, 20, 21), ]

model_T1 <- T1(unit_space_data = stackloss_center,
               signal_space_data = stackloss_signal,
               subtracts_V_e = TRUE,
               includes_transformed_data = TRUE)

# The following test samples are chosen casually.
stackloss_test <- stackloss[c(2, 12, 19), -4]

forecasting_T1 <- forecasting(model = model_T1,
                              newdata = stackloss_test,
                              includes_transformed_newdata = TRUE)
```

```
(forecasting_T1$y_hat) # Estimated values
(stackloss[c(2, 12, 19), 4]) # True values
```

---

forecasting.Ta *Forecasting method for the Ta method*

Description

forecasting.Ta (via forecasting) estimates the dependent values based on the Ta model.

Usage

```
## S3 method for class 'Ta'
forecasting(model, newdata, includes_transformed_newdata = FALSE)
```

Arguments

model
Object of class "Ta" generated by Ta or generates_model(..., method = "Ta").

newdata
Matrix with n rows (samples) and p columns (variables). The Data to be estimated. All data should be continuous values and should not have missing values.

includes_transformed_newdata
If TRUE, then the transformed data for newdata are included in a return object.

Value

A list containing the following components is returned.

M_hat
Vector with length n. The estimated values of the dependent variable after the data transformation.

y_hat
Vector with length n. The estimated values after the inverse transformation from M_hat.

model
Object of class "Ta" passed by model.

n
The number of samples for newdata.

q
The number of variables after the data transformation. q equals p.

X
If includes_transformed_newdata is TRUE, then the transformed data for newdata are included.

References

Inou, A., Nagata, Y., Horita, K., & Mori, A. (2012). Prediciton Accuracies of Improved Taguchi's T Methods Compared to those of Multiple Regresssion Analysis. *Journal of the Japanese Society for Quality Control, 42*(2), 103-115. (In Japanese)

Kawada, H., & Nagata, Y. (2015). An application of a generalized inverse regression estimator to Taguchi's T-Method. *Total Quality Science, 1*(1), 12-21.

See Also

general_forecasting.T and Ta

Examples

```
model_Ta <- Ta(sample_data = stackloss[-c(2, 12, 19), ],
               subtracts_V_e = TRUE,
               includes_transformed_data = TRUE)

forecasting_Ta <- forecasting(model = model_Ta,
                              newdata = stackloss[c(2, 12, 19), -4],
                              includes_transformed_newdata = TRUE)

(forecasting_Ta$y_hat) # Estimated values
(stackloss[c(2, 12, 19), 4]) # True values
```

---

forecasting.Tb   *Forecasting method for the Tb method*

Description

forecasting.Tb (via forecasting) estimates the dependent values based on the Tb model.

Usage

```
## S3 method for class 'Tb'
forecasting(model, newdata, includes_transformed_newdata = FALSE)
```

Arguments

model	Object of class "Tb" generated by Tb or generates_model(..., method = "Tb").
newdata	Matrix with n rows (samples) and p columns (variables). The Data to be estimated. All data should be continuous values and should not have missing values.
includes_transformed_newdata	If TRUE, then the transformed data for newdata are included in a return object.

Value

A list containing the following components is returned.

M_hat	Vector with length n. The estimated values of the dependent variable after the data transformation.
y_hat	Vector with length n. The estimated values after the inverse transformation from M_hat.
model	Object of class "Tb" passed by model.
n	The number of samples for newdata.

q	The number of variables after the data transformation. q equals p.
X	If `includes_transformed_newdata` is `TRUE`, then the transformed data for `newdata` are included.

References

Inou, A., Nagata, Y., Horita, K., & Mori, A. (2012). Prediciton Accuracies of Improved Taguchi's T Methods Compared to those of Multiple Regresssion Analysis. *Journal of the Japanese Society for Quality Control, 42*(2), 103-115. (In Japanese)

Kawada, H., & Nagata, Y. (2015). An application of a generalized inverse regression estimator to Taguchi's T-Method. *Total Quality Science, 1*(1), 12-21.

See Also

`general_forecasting.T` and `Tb`

Examples

```
model_Tb <- Tb(sample_data = stackloss[-c(2, 12, 19), ],
               subtracts_V_e = TRUE,
               includes_transformed_data = TRUE)

forecasting_Tb <- forecasting(model = model_Tb,
                              newdata = stackloss[c(2, 12, 19), -4],
                              includes_transformed_newdata = TRUE)

(forecasting_Tb$y_hat) # Estimated values
(stackloss[c(2, 12, 19), 4]) # True values
```

`general_diagnosis.MT` *General function to implement a diagnosis method for a family of Mahalanobis-Taguchi (MT) methods*

Description

`general_diagnosis.MT` is the general function that implements a diagnosis method for a family of Mahalanobis-Taguchi (MT) methods. Each diagnosis method of a family of MT methods can be implemented by setting the parameters of this function appropriately.

Usage

```
general_diagnosis.MT(unit_space, newdata, threshold,
  includes_transformed_newdata = FALSE)
```

Arguments

unit_space
: Object generated as a unit space.

newdata
: Matrix with n rows (samples) and p columns (variables). The data are used to calculate the desired distances from the unit space. All data should be continuous values and should not have missing values.

threshold
: Numeric specifying the threshold value to classify each sample into positive (TRUE) or negative (FALSE).

includes_transformed_newdata
: If TRUE, then the transformed data for newdata are included in a return object.

Value

A list containing the following components is returned.

distance
: Vector with length n. Distances from the unit space to each sample.

le_threshold
: Vector with length n. Logical values indicating the distance of each sample is less than or equal to the threhold value (TRUE) or not (FALSE).

threshold
: Numeric value to classify the sample into positive or negative.

unit_space
: Object passed by unit_space.

n
: The number of samples for newdata.

q
: The number of independent variables after the data transformation. According to the data transoformation function, q may be equal to p.

x
: If includes_transformed_newdata is TRUE, then the transformed data for newdata are included.

See Also

diagnosis.MT, diagnosis.MTA, and diagnosis.RT

Examples

```
# 40 data for versicolor in the iris dataset
iris_versicolor <- iris[61:100, -5]

# The following settings are same as the MT method.
unit_space <- general_MT(unit_space_data = iris_versicolor,
                         generates_transform_function =
                                            generates_normalization_function,
                         calc_A = function(x) solve(cor(x)),
                         includes_transformed_data = TRUE)

# 10 data for each kind (setosa, versicolor, virginica) in the iris dataset
iris_test <- iris[c(1:10, 51:60, 101:111), -5]

diagnosis <- general_diagnosis.MT(unit_space = unit_space,
                                  newdata = iris_test,
                                  threshold = 4,
```

```
                                    includes_transformed_newdata = TRUE)

(diagnosis$distance)
(diagnosis$le_threshold)
```

`general_forecasting.T` *General function to implement a forecasting method for a family of Taguchi (T) methods*

Description

`general_forecasting.T` is the general function that implements a forecasting method for a family of Taguchi (T) methods. Each forecasting method of a family of T methods can be implemented by setting the parameters of this function appropriately.

Usage

```
general_forecasting.T(model, newdata, includes_transformed_newdata = FALSE)
```

Arguments

`model`	Object generated as a model.
`newdata`	Matrix with n rows (samples) and p columns (variables). The data are used to calculate the desired distances from the unit space. All data should be continuous values and should not have missing values.
`includes_transformed_newdata`	If `TRUE`, then the transformed data for `newdata` are included in a return object.

Value

A list containing the following components is returned.

`M_hat`	Vector with length n. The estimated values of the dependent variable after the data trasformation.
`y_hat`	Vector with length n. The estimated values after the inverse transformation from `M_hat`.
`model`	Object passed by `model`.
`n`	The number of samples for `newdata`.
`q`	The number of variables after the data transformation.
`X`	If `includes_transformed_newdata` is `TRUE`, then the transformed data for `newdata` are included.

See Also

`forecasting.T1`, `forecasting.Ta`, and `forecasting.Tb`

Examples

```
# The value of the dependent variable of the following samples mediates
# in the stackloss dataset.
stackloss_center <- stackloss[c(9, 10, 11, 20, 21), ]

# The following samples are data other than the unit space data and the test
# data.
stackloss_signal <- stackloss[-c(2, 9, 10, 11, 12, 19, 20, 21), ]

# The following settings are same as the T1 method.
model <- general_T(unit_space_data = stackloss_center,
                   signal_space_data = stackloss_signal,
                   generates_transform_functions =
                                       generates_transformation_functions_T1,
                   subtracts_V_e = TRUE,
                   includes_transformed_data = TRUE)

# The following test samples are chosen casually.
stackloss_test <- stackloss[c(2, 12, 19), -4]

forecasting <- general_forecasting.T(model = model,
                                     newdata = stackloss_test,
                                     includes_transformed_newdata = TRUE)

(forecasting$y_hat) # Estimated values
(stackloss[c(2, 12, 19), 4]) # True values
```

---

general_MT — *General function to generate a unit space for a family of Mahalanobis-Taguchi (MT) methods*

Description

`general_MT` is a (higher-order) general function that generates a unit space for a family of Mahalanobis-Taguchi (MT) methods. Each MT method can be implemented by setting the parameters of this function appropriately.

Usage

```
general_MT(unit_space_data, calc_A, generates_transform_function,
  includes_transformed_data = FALSE)
```

Arguments

`unit_space_data`
: Matrix with n rows (samples) and p columns (variables). Data to generate the unit space. All data should be continuous values and should not have missing values.

`calc_A` Function that returns A in a quadratic form x'Ax. `calc_A` takes the transformed data as an (only) argument.

`generates_transform_function`
: Function that takes `unit_space_data` as an (only) argument and returns a data transformation function. The data transformation function takes data as an (only) argument and returns the transformed data.

`includes_transformed_data`
: If `TRUE`, then the transformed data are included in a return object.

Value

A list containing the following components is returned.

`A` q x q matrix calculated by `calc_A`.

`calc_A` Function passed by `calc_A`.

`transforms_data`
: Data transformation function generated from `generates_transform_function` based on `unit_space_data`.

`distance` Vector with length n. Distances from the unit space to each sample.

`n` The number of samples.

`q` The number of independent variables after the data transformation. According to the data transoformation function, q may be equal to p.

`x` If `includes_transformed_data` is `TRUE`, then the transformed data are included.

See Also

`MT`, `MTA` and `RT`

Examples

```
# 40 data for versicolor in the iris dataset
iris_versicolor <- iris[61:100, -5]

# The following settings are same as the MT method.
unit_space <- general_MT(unit_space_data = iris_versicolor,
                         generates_transform_function =
                                             generates_normalization_function,
                         calc_A = function(x) solve(cor(x)),
                         includes_transformed_data = TRUE)

(unit_space$distance)
```

general_T — *General function to generate a prediction expression for a family of Taguchi (T) methods*

Description

general_T is a (higher-order) general function that generates a prediction expression for a family of Taguchi (T) methods. Each T method can be implemented by setting the parameters of this function appropriately.

Usage

```
general_T(unit_space_data, signal_space_data, generates_transform_functions,
  subtracts_V_e = TRUE, includes_transformed_data = FALSE)
```

Arguments

unit_space_data
: Matrix with n rows (samples) and (p + 1) columns (variables). The 1 ~ p th columns are independent variables and the (p + 1) th column is a dependent variable. Underlying data to obtain a representative point for the normalization of the signal_space_data. All data should be continuous values and should not have missing values.

signal_space_data
: Matrix with m rows (samples) and (p + 1) columns (variables). The 1 ~ p th columns are independent variables and the (p + 1) th column is a dependent variable. Underlying data to generate a prediction expression. All data should be continuous values and should not have missing values.

generates_transform_functions
: A function that takes the unit_space_data as an (only) argument and returns a list containing three functions. A data transformation function for independent variables is the first component, a data transformation function for a dependent variable is the second component, and an inverse function of the data transformation function for a dependent variable is the third component. The data transformation function for independent variables takes independent variable data (a matrix of p columns) as an (only) argument and returns the transformed independent variable data. The data transformation function for a dependent variable takes dependent variable data (a vector) as an (only) argument and returns the transformed dependent variable data. The inverse function of the data transformation for a dependent variable takes the transformed dependent variable data (a vector) as an (only) argument and returns the untransformed dependent variable data.

subtracts_V_e
: If TRUE, then the error variance is subtracted in the numerator when calculating eta_hat.

includes_transformed_data
: If TRUE, then the transformed data are included in a return object.

Value

A list containing the following components is returned.

- `beta_hat` — Vector with length q. Estimated proportionality constants between each independent variable and the dependent variable.
- `subtracts_V_e` — Logical. If `TRUE`, then `eta_hat` was calculated without subtracting the error variance in the numerator.
- `eta_hat` — Vector with length q. Estimated squared signal-to-noise ratios (S/N) coresponding to `beta_hat`.
- `M_hat` — Vector with length n. The estimated values of the dependent variable after the data transformation for `signal_space_data`.
- `overall_prediction_eta` — Numeric. The overall squared signal-to-noise ratio (S/N).
- `transforms_independent_data` — Data transformation function generated from `generates_transform_functions` based on `unit_space_data`. The function for independent variables takes independent variable data (a matrix of p columns) as an (only) argument and returns the transformed independent variable data.
- `transforms_dependent_data` — Data transformation function generated in `generates_transform_functions` based on the `unit_space_data`. The function for a dependent variable takes dependent variable data (a vector) as an (only) argument and returns the transformed dependent variable data.
- `inverses_transformed_dependent_data` — Inverse function generated in the `generates_transform_functions` based on `unit_space_data`. The function of the takes the transformed dependent variable data (a vector) as an (only) argument and returns the dependent variable data inversed from the transformed dependent variable data.
- `m` — The number of samples for `signal_space_data`.
- `q` — The number of independent variables after the data transformation. According to the data transoformation function, q may be equal to p.
- `X` — If `includes_transformed_data` is `TRUE`, then the independent variable data after the data transformation for the `signal_space_data` are included.
- `M` — If `includes_transformed_data` is `TRUE`, then the (true) value of the dependent variable after the data transformation for the `signal_space_data` are included.

See Also

`T1`, `Ta`, and `Tb`

Examples

```
# The value of the dependent variable of the following samples mediates
# in the stackloss dataset.
stackloss_center <- stackloss[c(9, 10, 11, 20, 21), ]
```

```
# The following samples are data other than the unit space data and the test
# data.
stackloss_signal <- stackloss[-c(2, 9, 10, 11, 12, 19, 20, 21), ]

# The following settings are same as the T1 method.
model <- general_T(unit_space_data = stackloss_center,
                   signal_space_data = stackloss_signal,
                   generates_transform_functions =
                                   generates_transformation_functions_T1,
                   subtracts_V_e = TRUE,
                   includes_transformed_data = TRUE)

(model$M_hat)
```

---

generates_dimensionality_reduction_function
Function to generate a data transformation function for the Recognition-Taguchi (RT) method

---

Description

`generates_dimensionality_reduction_function` returns the data transformation function for the Recognition-Taguchi (RT) method based on the `unit_space_data`. The function reduces the dimensionality of data into 2 synthetic variables.

Usage

```
generates_dimensionality_reduction_function(unit_space_data)
```

Arguments

`unit_space_data`
: Matrix with n rows (samples) and p columns (variables). Data to generate the unit space. All data should be continuous values and should not have missing values.

Value

Function is returned which takes an n x p matrix as an (only) argument and returns a dimensionality-reduced n x 2 data frame with named columns; Y_1 and Y_2.

References

Taguchi, G. (2006). Objective Function and Generic Function (11). *Journal of Quality Engineering Society, 14*(2), 5-9. (In Japanese)

Huda, F., Kajiwara, I., Hosoya, N., & Kawamura, S. (2013). Bolt loosening analysis and diagnosis by non-contact laser excitation vibration tests. *Mechanical systems and signal processing, 40*(2), 589-604.

See Also

RT

Examples

```
# 40 data for versicolor in the iris dataset
iris_versicolor <- iris[61:100, -5]

reduces_dimensionality <-
                 generates_dimensionality_reduction_function(iris_versicolor)

is.function(reduces_dimensionality) # TRUE
```

---

generates_model — *Wrapper function to generate a model for a family of Taguchi (T) methods*

Description

`generates_model` generates a model for a family of Taguchi (MT) methods. The model of T1 method, Ta method or the Tb method can be generated by passing a method name (character) into a parameter `method`.

Usage

```
generates_model(unit_space_data, signal_space_data, sample_data,
  method = c("T1", "Ta", "Tb"), subtracts_V_e = TRUE,
  includes_transformed_data = FALSE)
```

Arguments

`unit_space_data`
: Used only for the T1 method. Matrix with n rows (samples) and (p + 1) columns (variables). The 1 ~ p th columns are independent variables and the (p + 1) th column is a dependent variable. Underlying data to obtain a representative point for the normalization of `signal_space_data`. All data should be continuous values and should not have missing values.

`signal_space_data`
: Used only for the T1 method. Matrix with m rows (samples) and (p + 1) columns (variables). The 1 ~ p th columns are independent variables and the (p + 1) th column is a dependent variable. Underlying data to generate a prediction expression. All data should be continuous values and should not have missing values.

`sample_data`
: Used for the Ta and the Tb methods. Matrix with n rows (samples) and (p + 1) columns (variables). The 1 ~ p th columns are independent variables and the (p + 1) th column is a dependent variable. All data should be continuous values and should not have missing values.

method	Character to designate a method. Currently, "MT", "MTA", and "RT" are available.
subtracts_V_e	If TRUE, then the error variance is subtracted in the numerator when calculating eta_hat.
includes_transformed_data	If TRUE, then the transformed data are included in a return object.

Value

A returned object depends on the selected method. See T1, Ta or Tb.

See Also

T1, Ta, Tb

Examples

```
# The value of the dependent variable of the following samples mediates
# in the stackloss dataset.
stackloss_center <- stackloss[c(9, 10, 11, 20, 21), ]

# The following samples are data other than the unit space data and the test
# data.
stackloss_signal <- stackloss[-c(2, 9, 10, 11, 12, 19, 20, 21), ]

# The following test samples are chosen casually.
stackloss_test <- stackloss[c(2, 12, 19), -4]

# T1 method
model_T1 <- generates_model(unit_space_data = stackloss_center,
                            signal_space_data = stackloss_signal,
                            method = "T1",
                            subtracts_V_e = TRUE)

forecasting_T1 <- forecasting(model = model_T1,
                              newdata = stackloss_test)

(forecasting_T1$y_hat)

# Ta method
model_Ta <- generates_model(sample_data =
                                   rbind(stackloss_center, stackloss_signal),
                            method = "Ta",
                            subtracts_V_e = TRUE)

forecasting_Ta <- forecasting(model = model_Ta,
                              newdata = stackloss_test)

(forecasting_Ta$y_hat)

# Tb method
model_Tb <- generates_model(sample_data =
```

```
                                rbind(stackloss_center, stackloss_signal),
                          method = "Tb",
                          subtracts_V_e = TRUE)

forecasting_Tb <- forecasting(model = model_Tb,
                            newdata = stackloss_test)

(forecasting_Tb$y_hat)
```

---

generates_normalization_function

Function to generate the data normalization function

Description

generates_normalization_function returns the data normalization function. The data normalization function is generated based on unit_space_data.

Usage

```
generates_normalization_function(unit_space_data, unit_space_center,
  unit_space_scale, is_scaled = TRUE)
```

Arguments

unit_space_data
: Matrix with n rows (samples) and p columns (variables). Data to generate the unit space. All data should be continuous values and should not have missing values.

unit_space_center
: Vector with length p. The values are subtrahends in normalization. If missing, the mean for each column of unit_space_data is used for normalization.

unit_space_scale
: Vector with length p. The values are divisors in normalization. If missing and is_scaled is TRUE, then the unbiased standard deviation for each column of unit_space_data is used for normalization.

is_scaled
: Logical. If TRUE (default value), normalization is conducted by subtracting unit_space_center and dividing by unit_space_scale. If FALSE, normalization is conducted by subtracting unit_space_center only.

Value

Function is returned which takes an n x p matrix as an (only) argument and returns a normalized n x p matrix. The normalization is conducted based on unit_space_data.

See Also

MT and MTA

Examples

```
# 40 data for versicolor in the iris dataset
iris_versicolor <- iris[61:100, -5]

normalizes_data <- generates_normalization_function(iris_versicolor)

is.function(normalizes_data) # TRUE
```

---

generates_transformation_functions_T1

Function to generate data transformation functions for the T1 methods

---

Description

generates_transformation_functions_T1 is the argument for the parameter generates_transform_functions in genera_T, which is used in the T1 method. In addtion, the Ta method also uses this function for the argument.

Usage

```
generates_transformation_functions_T1(unit_space_data)
```

Arguments

unit_space_data
: Matrix with n rows (samples) and (p + 1) columns (variables). Data to generate the unit space. All data should be continuous values and should not have missing values.

Value

generates_transformation_functions_T1 returns a list containing three functions. For the first component, the data transformation function for independent variables is a function that subtracts the mean of each independent variable. For the second component, the data transformation function for a dependent variable is a function that subtracts the mean of a dependent variable. For the third component, the inverse function of the data transformation function for a dependent variable is a function that adds the mean of a dependent variable. The mean used is the mean of the unit_space_data.

See Also

T1 and Ta

Examples

```
# The value of the dependent variable of the following samples mediates
# in the stackloss dataset.
stackloss_center <- stackloss[c(9, 10, 11, 20, 21), ]

tmp <- generates_transformation_functions_T1(stackloss_center)
mean_subtraction_function <- tmp[[1]]
subtracts_M_0 <- tmp[[2]]
adds_M_0 <- tmp[[3]]

is.function(mean_subtraction_function) # TRUE
is.function(subtracts_M_0) # TRUE
is.function(adds_M_0) # TRUE
```

---

generates_transformation_functions_Tb
Function to generate data transformation functions for the Tb methods

---

Description

generates_transformation_functions_Tb is the argument for the parameter generates_transform_functions in genera_T, which is used in the Tb method.

Usage

```
generates_transformation_functions_Tb(sample_data)
```

Arguments

sample_data	Matrix with n rows (samples) and (p + 1) columns (variables). The Tb method uses all data to generate the unit space. All data should be continuous values and should not have missing values.

Value

generates_transformation_functions_Tb returns a list containing three functions. For the first component, the data transformation function for independent variables is a function that subtracts the center of each independent variable. The center is determined in a specific manner for the Tb method. The center consists of each sample value which maximizes the signal-to-noise ratio (S/N) per independent variable. The values are determined independently so that different samples may be selected for different variables. For the second component, the data transformation function for a dependent variable is a function that subtracts the dependent variable of the sample which maximizes the S/N per independent variable. For the third component, the inverse function of the data transformation function for a dependent variable is a function that adds the weighted mean of a dependent variable. The weighted mean is calculated based on the S/N and the frequency of being selected in independent variables.

References

Inou, A., Nagata, Y., Horita, K., & Mori, A. (2012). Prediciton Accuracies of Improved Taguchi's T Methods Compared to those of Multiple Regresssion Analysis. *Journal of the Japanese Society for Quality Control, 42*(2), 103-115. (In Japanese)

Kawada, H., & Nagata, Y. (2015). An application of a generalized inverse regression estimator to Taguchi's T-Method. *Total Quality Science, 1*(1), 12-21.

See Also

Tb

Examples

```
# The value of the dependent variable of the following samples mediates
# in the stackloss dataset.
stackloss_center <- stackloss[c(9, 10, 11, 20, 21), ]

tmp <- generates_transformation_functions_Tb(stackloss_center)
center_subtraction_function <- tmp[[1]]
subtracts_ys <- tmp[[2]]
adds_M_0 <- tmp[[3]]

is.function(center_subtraction_function) # TRUE
is.function(subtracts_ys) # TRUE
is.function(adds_M_0) # TRUE
```

---

generates_unit_space *Wrapper function to generate a unit space for a family of Mahalanobis-Taguchi (MT) methods*

Description

`generates_unit_space` generates a unit space for a family of Mahalanobis-Taguchi (MT) methods. The unit space of MT method, MTA method or RT method can be generated by passing a method name (character) into a parameter `method`.

Usage

```
generates_unit_space(unit_space_data, method = c("MT", "MTA", "RT"),
  includes_transformed_data = FALSE, ...)
```

Arguments

`unit_space_data`
: Matrix with n rows (samples) and p columns (variables). Data to generate the unit space. All data should be continuous values and should not have missing values.

method — Character to designate a method. Currently, "MT", "MTA", and "RT" are available.

includes_transformed_data — If TRUE, then the transformed data are included in a return object.

... — Passed to solve for computing the inverse of the correlation matrix in MT and RT method.

Value

A returned object depends on the selected method. See MT, MTA or RT.

See Also

MT, MTA, RT, and solve

Examples

```
# 40 data for versicolor in the iris dataset
iris_versicolor <- iris[61:100, -5]

# 10 data for each kind (setosa, versicolor, virginica) in the iris dataset
iris_test <- iris[c(1:10, 51:60, 101:111), -5]

# MT method
unit_space_MT <- generates_unit_space(unit_space_data = iris_versicolor,
                                      method = "MT")

diagnosis_MT <- diagnosis(unit_space = unit_space_MT,
                          newdata = iris_test,
                          threshold = 4)

(diagnosis_MT$distance)
(diagnosis_MT$le_threshold)

# MTA method
unit_space_MTA <- generates_unit_space(unit_space_data = iris_versicolor,
                                       method = "MTA")

diagnosis_MTA <- diagnosis(unit_space = unit_space_MTA,
                           newdata = iris_test,
                           threshold = 0.5)

(diagnosis_MTA$distance)
(diagnosis_MTA$le_threshold)

# RT method
unit_space_RT <- generates_unit_space(unit_space_data = iris_versicolor,
                                      method = "RT")

diagnosis_RT <- diagnosis(unit_space = unit_space_RT,
                          newdata = iris_test,
                          threshold = 0.2)
```

```
(diagnosis_RT$distance)
(diagnosis_RT$le_threshold)
```

MT   *Function to generate a unit space for the Mahalanobis-Taguchi (MT) method*

Description

MT generates a unit space for the Mahalanobis-Taguchi (MT) method. In general_MT, the inversed correlation matrix is used for A and the data are normalized based on unit_space_data.

Usage

```
MT(unit_space_data, includes_transformed_data = FALSE, ...)
```

Arguments

unit_space_data
: Matrix with n rows (samples) and p columns (variables). Data to generate the unit space. All data should be continuous values and should not have missing values.

includes_transformed_data
: If TRUE, then the transformed data are included in a return object.

...
: Passed to solve for computing the inverse of the correlation matrix.

Value

MT returns an object of S3 class "MT". An object of class "MT" is a list containing the following components:

A	p x p (q x q) matrix. Inversed correlation matrix of unit_space_data (the transformed data).
calc_A	function(x) solve(cor(x), ...).
transforms_data	Function to be generated from generates_normalization_function based on unit_space_data.
distance	Vector with length n. Distances from the unit space to each sample.
n	The number of samples.
q	The number of variables after the data transformation. q is equal to p.
x	If includes_transformed_data is TRUE, then the transformed data are included.

References

Taguchi, G. (1995). Pattern Recognition and Quality Engineering (1). *Journal of Quality Engineering Society, 3*(2), 2-5. (In Japanese)

Taguchi, G., Wu, Y., & Chodhury, S. (2000). *Mahalanobis-Taguchi System.* McGraw-Hill Professional.

Taguchi, G., & Jugulum, R. (2002). *The Mahalanobis-Taguchi strategy: A pattern technology system.* John Wiley & Sons.

Woodall, W. H., Koudelik, R., Tsui, K. L., Kim, S. B., Stoumbos, Z. G., & Carvounis, C. P. (2003). A review and analysis of the Mahalanobis-Taguchi system. *Technometrics, 45*(1), 1-15.

See Also

solve, general_MT, generates_normalization_function, and diagnosis.MT

Examples

```
# 40 data for versicolor in the iris dataset
iris_versicolor <- iris[61:100, -5]

unit_space_MT <- MT(unit_space_data = iris_versicolor,
                    includes_transformed_data = TRUE)

# The following tol is a parameter passed to solve function.
unit_space_MT <- MT(unit_space_data = iris_versicolor,
                    includes_transformed_data = TRUE,
                    tol = 1e-9)

(unit_space_MT$distance)
```

MTA — *Function to generate a unit space for the Mahalanobis-Taguchi Adjoint (MTA) method*

Description

MTA generates a unit space for the Mahalanobis-Taguchi Adjoint (MTA) method. In general_MT, cofactor matrix is used for A and the data are normalized based on unit_space_data.

Usage

```
MTA(unit_space_data, includes_transformed_data = FALSE)
```

Arguments

`unit_space_data`
: Matrix with n rows (samples) and p columns (variables). Data to generate the unit space. All data should be continuous values and should not have missing values.

`includes_transformed_data`
: If `TRUE`, then the transformed data are included in a return object.

Value

`MTA` returns an object of S3 class "MTA". An object of class "MTA" is a list containing the following components:

`A`	p x p (q x q) matrix. Cofactor matrix of `unit_space_data` (the transformed data).
`calc_A`	`calc_cofactor`.
`transforms_data`	Function to be generated from the `generates_normalization_function` based on the `unit_space_data`.
`distance`	Vector with length n. Distances from the unit space to each sample.
`n`	The number of samples.
`q`	The number of variables after the data transformation. q equals p.
`x`	If `includes_transformed_data` is `TRUE`, then the transformed data are included.

References

Taguchi, G. & Kanetaka, T. (2002). *Engineering Technical Development in MT System - Lecture on Applied Quality.* Japanese Standards Association. (In Japanese)

Taguchi, G., & Jugulum, R. (2002). *The Mahalanobis-Taguchi strategy: A pattern technology system.* John Wiley & Sons.

See Also

`calc_cofactor`, `general_MT`, `generates_normalization_function`, and `diagnosis.MT`

Examples

```
# 40 data for versicolor in the iris dataset
iris_versicolor <- iris[61:100, -5]

unit_space_MTA <- MTA(unit_space_data = iris_versicolor,
                      includes_transformed_data = TRUE)

(unit_space_MTA$distance)
```

RT *Function to generate a unit space for the Recognition-Taguchi (RT) method*

Description

RT generates a unit space for the Recognition-Taguchi (RT) method. In general_MT, the inversed correlation matrix is used for A and the data are transformed by the function to be generated by generates_dimensionality_reduction_function based on unit_space_data. In the transformation, the p variables in unit_space_data are reduced into 2 synthetic variables.

Usage

```
RT(unit_space_data, includes_transformed_data = FALSE, ...)
```

Arguments

unit_space_data
: Matrix with n rows (samples) and p columns (variables). Data to generate the unit space. All data should be continuous values and should not have missing values.

includes_transformed_data
: If TRUE, then the transformed data are included in a return object.

...
: Passed to solve for computing the inverse of the correlation matrix.

Value

RT returns an object of S3 class "RT". An object of class "RT" is a list containing the following components:

A
: 2 x 2 matrix. Inversed correlation matrix of the transformed unit_space_data.

calc_A
: function(x) solve(cor(x), ...).

transforms_data
: Function to be generated from generates_dimensionality_reduction_function based on unit_space_data.

distance
: Vector with length n. Distances from the unit space to each sample.

n
: The number of samples.

q
: The number of variables after the data transformation. q is always 2.

x
: If includes_transformed_data is TRUE, then the transformed data are included.

References

Taguchi, G. (2006). Objective Function and Generic Function (11). *Journal of Quality Engineering Society, 14*(2), 5-9. (In Japanese)

Huda, F., Kajiwara, I., Hosoya, N., & Kawamura, S. (2013). Bolt loosening analysis and diagnosis by non-contact laser excitation vibration tests. *Mechanical systems and signal processing, 40*(2), 589-604.

See Also

`solve`, `general_MT`, `generates_dimensionality_reduction_function`, and `diagnosis.MT`

Examples

```
# 40 data for versicolor in the iris dataset
iris_versicolor <- iris[61:100, -5]

unit_space_RT <- RT(unit_space_data = iris_versicolor,
                    includes_transformed_data = TRUE)

# The following "tol" is a parameter passed to the solve function.
unit_space_RT <- RT(unit_space_data = iris_versicolor,
                    includes_transformed_data = TRUE,
                    tol = 1e-9)

(unit_space_RT$distance)
```

T1 — *Function to generate a prediction expression for the two-sided Taguchi (T1) method*

Description

T1 generates a prediction expression for the two-sided Taguchi (T1) method. In `general_T`, the data are normalized by subtracting the mean and without scaling based on `unit_space_data`. The sample data should be divided into 2 datasets in advance. One is for the unit space and the other is for the signal space.

Usage

```
T1(unit_space_data, signal_space_data, subtracts_V_e = TRUE,
  includes_transformed_data = FALSE)
```

Arguments

`unit_space_data`
: Matrix with n rows (samples) and (p + 1) columns (variables). The 1 ~ p th columns are independent variables and the (p + 1) th column is a dependent variable. Underlying data to obtain a representative point for the normalization of the `signal_space_data`. All data should be continuous values and should not have missing values.

`signal_space_data`
: Matrix with m rows (samples) and (p + 1) columns (variables). The 1 ~ p th columns are independent variables and the (p + 1) th column is a dependent variable. Underlying data to generate a prediction expression. All data should be continuous values and should not have missing values.

subtracts_V_e If TRUE, then the error variance is subtracted in the numerator when calculating eta_hat.

includes_transformed_data
If TRUE, then the transformed data are included in a return object.

Value

A list containing the following components is returned.

beta_hat Vector with length q. Estimated proportionality constants between each independent variable and the dependent variable.

subtracts_V_e Logical. If TRUE, then eta_hat was calculated without subtracting the error variance in the numerator.

eta_hat Vector with length q. Estimated squared signal-to-noise ratios (S/N) coresponding to beta_hat.

M_hat Vector with length n. The estimated values of the dependent variable after the data transformation for signal_space_data.

overall_prediction_eta
Numeric. The overall squared signal-to-noise ratio (S/N).

transforms_independent_data
Data transformation function generated from generates_transform_functions based on the unit_space_data. The function for independent variables takes independent variable data (a matrix of p columns) as an (only) argument and returns the transformed independent variable data.

transforms_dependent_data
Data transformation function generated from generates_transform_functions based on the unit_space_data. The function for a dependent variable takes dependent variable data (a vector) as an (only) argument and returns the transformed dependent variable data.

inverses_dependent_data
Data transformation function generated from generates_transform_functions based on the unit_space_data. The function of the takes the transformed dependent variable data (a vector) as an (only) argument and returns the dependent variable data inversed from the transformed dependent variable data.

m The number of samples for signal_space_data.

q The number of independent variables after the data transformation. q equals p.

X If includes_transformed_data is TRUE, then the independent variable data after the data transformation for the signal_space_data are included.

M If includes_transformed_data is TRUE, then the (true) value of the dependent variable after the data transformation for the signal_space_data are included.

References

Taguchi, G. (2006). Objective Function and Generic Function (12). *Journal of Quality Engineering Society, 14*(3), 5-9. (In Japanese)

Inou, A., Nagata, Y., Horita, K., & Mori, A. (2012). Prediciton Accuracies of Improved Taguchi's T Methods Compared to those of Multiple Regresssion Analysis. *Journal of the Japanese Society for Quality Control, 42*(2), 103-115. (In Japanese)

Kawada, H., & Nagata, Y. (2015). An application of a generalized inverse regression estimator to Taguchi's T-Method. *Total Quality Science, 1*(1), 12-21.

See Also

`general_T`, `generates_transformation_functions_T1`, and `forecasting.T1`

Examples

```
# The value of the dependent variable of the following samples mediates
# in the stackloss dataset.
stackloss_center <- stackloss[c(9, 10, 11, 20, 21), ]

# The following samples are data other than the unit space data and the test
# data.
stackloss_signal <- stackloss[-c(2, 9, 10, 11, 12, 19, 20, 21), ]

model_T1 <- T1(unit_space_data = stackloss_center,
               signal_space_data = stackloss_signal,
               subtracts_V_e = TRUE,
               includes_transformed_data = TRUE)

(model_T1$M_hat)
```

Ta — *Function to generate a prediction expression for the Ta method*

Description

`Ta` generates a prediction expression for the Ta method. In `general_T`, the data are normalized by subtracting the mean and without scaling based on `sample_data`. The sample data are not divided into 2 datasets. All the sample data are used for both unit space and signal space.

Usage

```
Ta(sample_data, subtracts_V_e = TRUE, includes_transformed_data = FALSE)
```

Arguments

`sample_data`	Matrix with n rows (samples) and (p + 1) columns (variables). The 1 ~ p th columns are independent variables and the (p + 1) th column is a dependent variable. All data should be continuous values and should not have missing values.
`subtracts_V_e`	If `TRUE`, then the error variance is subtracted in the numerator when calculating `eta_hat`.

includes_transformed_data
: If TRUE, then the transformed data are included in a return object.

Value

A list containing the following components is returned.

beta_hat
: Vector with length q. Estimated proportionality constants between each independent variable and the dependent variable.

subtracts_V_e
: Logical. If TRUE, then eta_hat was calculated without subtracting the error variance in the numerator.

eta_hat
: Vector with length q. Estimated squared signal-to-noise ratios (S/N) coresponding to beta_hat.

M_hat
: Vector with length n. The estimated values of the dependent variable after the data transformation for sample_data.

overall_prediction_eta
: Numeric. The overall squared signal-to-noise ratio (S/N).

transforms_independent_data
: Data transformation function generated from generates_transform_functions based on the unit_space_data. The function for independent variables takes independent variable data (a matrix of p columns) as an (only) argument and returns the transformed independent variable data.

transforms_dependent_data
: Data transformation function generated from generates_transform_functions based on the unit_space_data. The function for a dependent variable takes dependent variable data (a vector) as an (only) argument and returns the transformed dependent variable data.

inverses_dependent_data
: Data transformation function generated from generates_transform_functions based on the unit_space_data. The function of the takes the transformed dependent variable data (a vector) as an (only) argument and returns the dependent variable data inversed from the transformed dependent variable data.

m
: The number of samples for sample_data.

q
: The number of independent variables after the data transformation. q equals p.

X
: If includes_transformed_data is TRUE, then the independent variable data after the data transformation for the sample_data are included.

M
: If includes_transformed_data is TRUE, then the (true) value of the dependent variable after the data transformation for the sample_data are included.

References

Inou, A., Nagata, Y., Horita, K., & Mori, A. (2012). Prediciton Accuracies of Improved Taguchi's T Methods Compared to those of Multiple Regresssion Analysis. *Journal of the Japanese Society for Quality Control, 42*(2), 103-115. (In Japanese)

Kawada, H., & Nagata, Y. (2015). An application of a generalized inverse regression estimator to Taguchi's T-Method. *Total Quality Science, 1*(1), 12-21.

See Also

general_T, generates_transformation_functions_T1, and forecasting.Ta

Examples

```
model_Ta <- Ta(sample_data = stackloss[-c(2, 12, 19), ],
               subtracts_V_e = TRUE,
               includes_transformed_data = TRUE)

(model_Ta$M_hat)
```

Tb — *Function to generate a prediction expression for the Tb method*

Description

Tb generates a prediction expression for the Tb method. In general_T, the data are normalized by subtracting the center and without scaling based on sample_data. The center is determined by the specific way for the Tb method. For details, please see generates_transformation_functions_Tb. All the sample data are used for both unit space and signal space.

Usage

```
Tb(sample_data, subtracts_V_e = TRUE, includes_transformed_data = FALSE)
```

Arguments

sample_data	Matrix with n rows (samples) and (p + 1) columns (variables). The 1 ~ p th columns are independent variables and the (p + 1) th column is a dependent variable. All data should be continuous values and should not have missing values.
subtracts_V_e	If TRUE, then the error variance is subtracted in the numerator when calculating eta_hat.
includes_transformed_data	If TRUE, then the transformed data are included in a return object.

Value

A list containing the following components is returned.

beta_hat	Vector with length q. Estimated proportionality constants between each independent variable and the dependent variable.
subtracts_V_e	Logical. If TRUE, then eta_hat was calculated without subtracting the error variance in the numerator.
eta_hat	Vector with length q. Estimated squared signal-to-noise ratios (S/N) coresponding to beta_hat.

M_hat
: Vector with length n. The estimated values of the dependent variable after the data transformation for sample_data.

overall_prediction_eta
: Numeric. The overall squared signal-to-noise ratio (S/N).

transforms_independent_data
: Data transformation function generated from generates_transform_functions based on the unit_space_data. The function for independent variables takes independent variable data (a matrix of p columns) as an (only) argument and returns the transformed independent variable data.

transforms_dependent_data
: Data transformation function generated from generates_transform_functions based on the unit_space_data. The function for a dependent variable takes dependent variable data (a vector) as an (only) argument and returns the transformed dependent variable data.

inverses_dependent_data
: Data transformation function generated from generates_transform_functions based on the unit_space_data. The function of the takes the transformed dependent variable data (a vector) as an (only) argument and returns the dependent variable data inversed from the transformed dependent variable data.

m
: The number of samples for sample_data.

q
: The number of independent variables after the data transformation. q equals p.

X
: If includes_transformed_data is TRUE, then the independent variable data after the data transformation for the sample_data are included.

M
: If includes_transformed_data is TRUE, then the (true) value of the dependent variable after the data transformation for the sample_data are included.

References

Inou, A., Nagata, Y., Horita, K., & Mori, A. (2012). Prediciton Accuracies of Improved Taguchi's T Methods Compared to those of Multiple Regresssion Analysis. *Journal of the Japanese Society for Quality Control, 42*(2), 103-115. (In Japanese)

Kawada, H., & Nagata, Y. (2015). An application of a generalized inverse regression estimator to Taguchi's T-Method. *Total Quality Science, 1*(1), 12-21.

See Also

general_T, generates_transformation_functions_Tb, and forecasting.Tb

Examples

```
model_Tb <- Tb(sample_data = stackloss[-c(2, 12, 19), ],
               subtracts_V_e = TRUE,
               includes_transformed_data = TRUE)

(model_Tb$M_hat)
```

Index

calc_cofactor, 2, *33*
calc_M_hat, 3
calc_overall_predicton_eta, 4
class, *31*, *33, 34*

diagnosis, 5, *6–10*
diagnosis.MT, *5, 6*, 6, *17*, *32, 33*, *35*
diagnosis.MTA, *5, 6*, 8, *17*
diagnosis.RT, *5, 6*, 10, *17*

forecasting, 11, *12*, *14, 15*
forecasting.T1, *11, 12*, 12, *18*, *37*
forecasting.Ta, *11, 12*, 14, *18*, *39*
forecasting.Tb, *11, 12*, 15, *18*, *40*

general_diagnosis.MT, *5*, *7*, *9*, *11*, 16
general_forecasting.T, *3*, *5*, *11*, *13*, *15, 16*, 18
general_MT, 19, *31–35*
general_T, *3*, *5*, 21, *35*, *37*, *39, 40*
generates_dimensionality_reduction_function, 23, *34, 35*
generates_model, *12*, *14, 15*, 24
generates_normalization_function, 26, *31–33*
generates_transformation_functions_T1, 27, *37*, *39*
generates_transformation_functions_Tb, 28, *39, 40*
generates_unit_space, *6–8*, *10*, 29

MT, *6, 7*, *20*, *26*, *29, 30*, 31
MTA, *3*, *8, 9*, *20*, *26*, *29, 30*, 32

RT, *10, 11*, *20*, *24*, *29, 30*, 34

solve, *30–32*, *34, 35*

T1, *12, 13*, *22*, *24, 25*, *27*, 35
Ta, *14, 15*, *22*, *24, 25*, *27*, 37
Tb, *15, 16*, *22*, *24, 25*, *29*, 39