Package 'badp'

March 11, 2026

Title Bayesian Averaging for Dynamic Panels

Version 0.4.0

Description Implements Bayesian model averaging for dynamic panels with weakly exogenous regressors as described in the paper by Moral-Benito (2013, <doi:10.1080/07350015.2013.818003>). The package provides functions to estimate dynamic panel data models and analyze the results of the estimation.

License MIT + file LICENSE

URL https://badp-project.github.io/bdsm/, https://github.com/badp-project/bdsm

BugReports https://github.com/badp-project/bdsm/issues

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Suggests pkgdown, rmarkdown, spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

Imports dplyr, ggplot2, ggpubr, grid, gridExtra, knitr, magrittr, optimbase, parallel, pbapply, Rcpp, RcppArmadillo, rje, rlang, rootSolve, stats, tidyr, tidyselect

LinkingTo Rcpp, RcppArmadillo

VignetteBuilder knitr

Depends R (>= 3.5)

Language en-US

NeedsCompilation yes

Author Krzysztof Beck [aut], Piotr Cukier [aut], Marcin Dubel [aut, cre], Mariusz Szczepanczyk [aut], Mateusz Wyszynski [aut]

Maintainer Marcin Dubel <marcindubel@gmail.com>

Repository CRAN

Date/Publication 2026-03-11 19:30:09 UTC

Contents

best_models *Table with the best models according to one of the posterior criteria*

Description

This function creates a ranking of best models according to one of the possible criterion (PMP under binomial model prior, PMP under binomial-beta model prior, R^2 under binomial model prior, R^2 under binomial-beta model prior). The function gives two types of tables in three different formats: inclusion table (where 1 indicates presence of the regressor in the model and 0 indicates that the variable is excluded from the model) and estimation results table (it displays the best models and estimation output for those models: point estimates, standard errors, significance level, and R^2).

Usage

```
best_models(
  bma_list,
  criterion = 1,
  best = 5,
  round = 3,
  estimate = TRUE,
  robust = TRUE
)
```

Arguments

bma_list	bma object (the result of the bma function)
criterion	The criterion that will be used for a basis of the model ranking: 1 - binomial model prior 2 - binomial-beta model prior
best	The number of the best models to be considered
round	Parameter indicating the decimal place to which number in the tables should be rounded (default round = 3)
estimate	A parameter with values TRUE or FALSE indicating which table should be displayed when TRUE - table with estimation to the results FALSE - table with the inclusion of regressors in the best models
robust	A parameter with values TRUE or FALSE indicating which type of stanrdard errors should be displayed when the function finishes calculations. Works only if estimate = TRUE. Works well when best is small. TRUE - robust standard errors FALSE - regular standard errors

Value

A list with best_models objects:

1. matrix with inclusion of the regressors in the best models
2. matrix with estimation output in the best models with regular standard errors
3. matrix with estimation output in the best models with robust standard errors
4. knitr_kable table with inclusion of the regressors in the best models (the best for the display on the console - up to 11 models)
5. knitr_kable table with estimation output in the best models with regular standard errors (the best for the display on the console - up to 6 models)
6. knitr_kable table with estimation output in the best models with robust standard errors (the best for the display on the console - up to 6 models)
7. gTree table with inclusion of the regressors in the best models (displayed as a plot). Use grid::grid.draw() to display.
8. gTree table with estimation output in the best models with regular standard errors (displayed as a plot). Use grid::grid.draw() to display.
9. gTree table with estimation output in the best models with robust standard errors (displayed as a plot). Use grid::grid.draw() to display.

Examples

```
library(magrittr)

data_prepared <- badp::economic_growth[, 1:6] %>%
  badp::feature_standardization(
    excluded_cols = c(country, year, gdp)
  ) %>%
  badp::feature_standardization(
    group_by_col  = year,
    excluded_cols = country,
    scale         = FALSE
  )

bma_results <- bma(
  model_space = badp::small_model_space,
  round       = 3,
  dilution    = 0
)

best_5_models <- best_models(bma_results, criterion = 1, best = 5, estimate = TRUE, robust = TRUE)
```

bma *Calculation of the bma object*

Description

This function calculates BMA statistics based on the provided model space. Other objects for furhter analysis are also returned.

Usage

```
bma(model_space, round = 4, EMS = NULL, dilution = 0, dil.Par = 0.5)
```

Arguments

model_space	List with params and stats from the model space
round	Parameter indicating the decimal place to which number in the BMA tables and prior and posterior model sizes should be rounded (default round = 4)
EMS	Expected model size for model binomial and binomial-beta model prior
dilution	Binary parameter: 0 - NO application of a dilution prior; 1 - application of a dilution prior (George 2010).
dil.Par	Parameter associated with dilution prior - the exponent of the determinant (George 2010). Used only if parameter dilution = 1.

Value

A list with 16 elements.

uniform_table A table containing the results based on the binomialmodel prior.

random_table A table containing the results based on the binomial-beta model prior.

reg_names A vector containing the names of the regressors, used by the functions.

R The total number of regressors.

num_of_models The number of models present in the model space.

forJointnes A table containing model IDs and posterior model probabilities (PMPs) for the jointness function.

forBestModels A table containing model IDs, PMPs, coefficients, standard deviations,and standardized regression coefficients (stdRs) for the best_models function.

EMS The expected model size for the binomial and binomial-beta model priors, as specified by the user (default is EMS = R/2).

sizePriors A table of uniform and random model priors distributed over model sizes for the model_sizes function.

PMPs A table containing the posterior model probabilities for use in the model_sizes function.

modelPriors A table containing the model priors, used by the model_pmp function.

dilution A parameter indicating whether the priors were diluted, used in the model_sizes function.

alphas A vector of coefficients for the lagged dependent variable in the coef_hist function.

betas_nonzero A vector of nonzero coefficients for the regressors in the coef_hist function.

d_free A table containing the degrees of freedom for the estimated models in the best_models function.

PMStable A table containing the prior and posterior expected model sizes for the binomial and binomial-beta model priors.

Examples

```
library(magrittr)

data_prepared <- badp::economic_growth[, 1:6] %>%
  badp::feature_standardization(
    excluded_cols = c(country, year, gdp)
  ) %>%
  badp::feature_standardization(
    group_by_col  = year,
    excluded_cols = country,
    scale         = FALSE
  )

bma_results <- bma(
  model_space = badp::small_model_space,
  round       = 3,
  dilution    = 0
)
```

`coef_hist` *Graphs of the distribution of the coefficients over the model space*

Description

This function draws graphs of the distribution (in the form of histogram or kernel density) of the coefficients for all the considered regressors over the part of the model space that includes this regressors (half of the model space).

Arguments

`bma_list`	bma object (the result of the bma function)
`weight`	Parameter indicating whether the coefficients should be weighted by posterior model probabilities: 1. NULL - no weighting (default option) 2. "binomial" - using posterior model probabilities based on binomial model prior

3. "beta" - using posterior model probabilities based on binomial-beta model prior

`BW` Parameter indicating what method should be chosen to find bin widths for the histograms:

1. "FD" Freedman-Diaconis method
2. "SC" Scott method
3. "vec" user specified bin widths provided through a vector (parameter: binW)

`binW` A vector with bin widths to be used to construct histograms for the regressors. The vector must be of the size equal to total number of regressors. The vector with bin widths is used only if parameter BW="vec".

`BN` Parameter taking the values (default: BN = 0):
1 - the histogram will be build based on the number of bins specified by the user through parameter num. If BN=1, the function ignores parameters BW.
0 - the histogram will be build in line with parameter BW

`num` A vector with the numbers of bins used to be used to construct histograms for the regressors. The vector must be of the size equal to total number of regressors. The vector with bin widths is used only if parameter BN=1.

`kernel` A parameter taking the values (default: kernel = 0):
1 - the function will build graphs using kernel density for the distribution of coefficients (with kernel=1, the function ignores parameters BW and BN)
0 - the function will build regular histogram density for the distribution of coefficients

Value

A list with the graphs of the distribution of coefficients for all the considered regressors.

Examples

```
library(magrittr)

data_prepared <- badp::economic_growth[, 1:6] %>%
  badp::feature_standardization(
    excluded_cols = c(country, year, gdp)
  ) %>%
  badp::feature_standardization(
    group_by_col  = year,
    excluded_cols = country,
    scale         = FALSE
  )

bma_results <- bma(
  model_space = badp::small_model_space,
  round       = 3,
  dilution    = 0
```

```
)

coef_plots <- coef_hist(bma_results, kernel = 1)
```

`compute_model_space_stats` *Approximate standard deviations for the models*

Description

Approximate standard deviations are computed for the models in the given model space. Two versions are computed.

Usage

```
compute_model_space_stats(
  df,
  dep_var_col,
  timestamp_col,
  entity_col,
  params,
  nested = TRUE,
  exact_value = FALSE,
  model_prior = "uniform",
  cl = NULL
)
```

Arguments

`df`	Data frame with data for the SEM analysis.
`dep_var_col`	Column with the dependent variable
`timestamp_col`	The name of the column with timestamps
`entity_col`	Column with entities (e.g. countries)
`params`	A matrix (with named rows) with each column corresponding to a model. Each column specifies model parameters. Compare with optim_model_space_params
`nested`	Logical. If `TRUE` (default), compute approximate standard deviations using the nested-model approach via `nested_std_dev_from_params()`. If `FALSE`, use the non-nested approach via `non_nested_std_dev_from_params()`. The choice affects which approximation routine is used for each model in `params`.
`exact_value`	Whether the exact value of the likelihood should be computed (`TRUE`) or just the proportional part (`FALSE`). Check sem_likelihood for details.
`model_prior`	Which model prior to use. For now there are two options: `'uniform'` and `'binomial-beta'`. Default is `'uniform'`.
`cl`	An optional cluster object. If supplied, the function will use this cluster for parallel processing. If `NULL` (the default), `pbapply::pblapply` will run sequentially.

Value

Matrix with columns describing likelihood and standard deviations for each model. The first row is the likelihood for the model (computed using the parameters in the provided model space). The second row is almost 1/2 * BIC_k as in Raftery's Bayesian Model Selection in Social Research eq. 19 (see TODO in the code below). The third row is model posterior probability. Then there are rows with standard deviations for each parameter. After that we have rows with robust standard deviation (not sure yet what exactly "robust" means).

Examples

```
library(magrittr)
data_prepared <- badp::economic_growth[, 1:6] %>%
  badp::feature_standardization(
    excluded_cols = c(country, year, gdp)
  ) %>%
  badp::feature_standardization(
    group_by_col  = year,
    excluded_cols = country,
    scale         = FALSE
  )

compute_model_space_stats(
  df            = data_prepared,
  dep_var_col   = gdp,
  timestamp_col = year,
  entity_col    = country,
  params        = small_model_space$params
)
```

---

economic_growth *Economic Growth Data*

Description

Data used in Growth Empirics in Panel Data under Model Uncertainty and Weak Exogeneity (Moral-Benito, 2016, Journal of Applied Econometrics).

Usage

```
economic_growth
```

Format

`economic_growth`**:**

A data frame with 365 rows and 12 columns (73 countries and 4 periods + extra one for lagged dependent variable):

year Year

country Country ID

gdp Logarithm of GDP per capita (2000 US dollars at PP)

ish Ratio of real domestic investment to GDP

sed Stock of years of secondary education in the total population

pgrw Average growth rate of population

pop Population in millions of people

ipr Purchasing-power-parity numbers for investment goods

opem Exports plus imports as a share of GDP

gsh Ratio of government consumption to GDP

lnlex Logarithm of the life expectancy at birth

polity Composite index given by the democracy score minus the autocracy score

Source

http://qed.econ.queensu.ca/jae/datasets/moral-benito001/

---

exogenous_matrix *Matrix with exogenous variables for SEM representation*

Description

Create matrix which contains exogenous variables used in the Simultaneous Equations Model (SEM) representation. Currently these are: dependent variable from the lowest time stamp and regressors from the second lowest time stamp. The matrix is then used to compute likelihood for SEM analysis.

Usage

```
exogenous_matrix(df, timestamp_col, entity_col, dep_var_col)
```

Arguments

df	Data frame with data for the SEM analysis.
timestamp_col	Column which determines time periods. For now only natural numbers can be used as timestamps
entity_col	Column which determines entities (e.g. countries, people)
dep_var_col	Column with dependent variable

Value

Matrix of size N x k+1 where N is the number of entities considered and k is the number of chosen regressors

Examples

```
set.seed(1)
df <- data.frame(
  entities = rep(1:4, 5),
  times = rep(seq(1960, 2000, 10), each = 4),
  dep_var = stats::rnorm(20), a = stats::rnorm(20), b = stats::rnorm(20)
)
exogenous_matrix(df, times, entities, dep_var)
```

`extract_names` *Extraction of names of the variables*

Description

The function extract the names of the variables from the data set used in the analysis and places them in a vector.

Usage

```
extract_names(df)
```

Arguments

`df`	Data frame with data for the analysis.

Value

A vector with names of the variables.

Examples

```
df <- badp::economic_growth

reg_names <- extract_names(df)
```

`feature_standardization` *Perform feature standardization*

Description

This function performs feature standardization (also known as z-score normalization) by centering the features around their mean and scaling by their standard deviation.

Usage

```
feature_standardization(df, excluded_cols, group_by_col, scale = TRUE)
```

Arguments

`df`	Data frame with the data.
`excluded_cols`	Unquoted column names to exclude from standardization. If missing, all columns are standardized.
`group_by_col`	Unquoted column names to group the data by before applying standardization. If missing, no grouping is performed.
`scale`	Logical. If `TRUE` (default) scales by the standard deviation.

Value

A data frame with standardized features.

Examples

```
df <- data.frame(
  year = c(2000, 2001, 2002, 2003, 2004),
  country = c("A", "A", "B", "B", "C"),
  gdp = c(1, 2, 3, 4, 5),
  ish = c(2, 3, 4, 5, 6),
  sed = c(3, 4, 5, 6, 7)
)

# Standardize every column
df_with_only_numeric_values <- df[, setdiff(names(df), "country")]
feature_standardization(df_with_only_numeric_values)

# Standardize all columns except 'country'
feature_standardization(df, excluded_cols = country)

# Standardize across countries (grouped by 'country')
feature_standardization(df, group_by_col = country)

# Standardize, excluding 'country' and group-wise by 'year'
feature_standardization(df, excluded_cols = country, group_by_col = year)
```

full_bma_results *Example output of the bma function*

Description

A list with multiple elements summarising the BMA analysis

Usage

```
full_bma_results
```

Format

An object of class `list` of length 16.

full_model_space *Example output of* `optim_model_space`

Description

A list created with `optim_model_space` using the `economic_growth` dataset.

Usage

```
full_model_space
```

Format

A list with 5 elements:

params A numeric matrix with 40 rows and 512 columns, containing parameter values for the full model space. Each column represents a different model.

stats A numeric matrix of statistics computed by `compute_model_space_stats` based on `params`. Row 1 contains model likelihoods. Row 2 contains a quantity proportional to `0.5 * BIC` (cf. Raftery, Bayesian Model Selection in Social Research, Eq. 19). Rows 3–7 contain standard deviations, and rows 8–12 contain robust standard deviations.

reg_names A character vector with the names of the variables.

observations_num The total number of observations in the panel (292).

df The data frame used in the analysis.

hessian *Hessian matrix*

Description

Creates the hessian matrix for a given likelihood function.

Usage

```
hessian(lik, theta, ...)
```

Arguments

lik	function
theta	kx1 matrix
...	other parameters passed to lik function.

Value

Hessian kxk matrix where k is the number of parameters included in the theta matrix

Examples

```
lik <- function(theta) {
 return(theta[1]^2 + theta[2]^2)
}

hessian(lik, c(1, 1))
```

init_model_space_params *Initialize model space matrix*

Description

This function builds a representation of the model space, by creating a dataframe where each column represents values of the parameters for a given model. Real value means that the parameter is included in the model. A parameter not present in the model is marked as NA.

Usage

```
init_model_space_params(
  df,
  timestamp_col,
  entity_col,
  dep_var_col,
  init_value = 1
)
```

Arguments

df	Data frame with data for the SEM analysis.
timestamp_col	Column which determines time periods. For now only natural numbers can be used as timestamps
entity_col	Column which determines entities (e.g. countries, people)
dep_var_col	Column with dependent variable
init_value	Initial value for parameters present in the model. Default is 1.

Details

Currently the set of features is assumed to be all columns which remain after excluding `timestamp_col`, `entity_col` and `dep_var_col`.

A power set of all possible exclusions of linear dependence on the given feature is created, i.e. if there are 4 features we end up with 2^4 possible models (for each model we independently decide whether to include or not a feature).

Value

matrix of model parameters

Examples

```
library(magrittr)

data_prepared <- badp::economic_growth[, 1:5] %>%
  badp::feature_standardization(
    excluded_cols = c(country, year, gdp)
  ) %>%
  badp::feature_standardization(
    group_by_col  = year,
    excluded_cols = country,
    scale         = FALSE
  )

init_model_space_params(data_prepared, year, country, gdp)
```

`jointness` *Calculation of of the jointness measures*

Description

This function calculates four types of the jointness measures based on the posterior model probabilities calculated using binomial and binomial-beta model prior. The four measures are:

1. HCGHM - for Hofmarcher et al. (2018) measure;
2. LS - for Ley & Steel (2007) measure;
3. DW - for Doppelhofer & Weeks (2009) measure;
4. PPI - for posterior probability of including both variables.
   The measures under binomial model prior will appear in a table above the diagonal, and the measure calculated under binomial-beta model prior below the diagonal.

   REFERENCES
   Doppelhofer G, Weeks M (2009) Jointness of growth determinants. Journal of Applied Econometrics., 24(2), 209-244. doi: 10.1002/jae.1046
   Hofmarcher P, Crespo Cuaresma J, Grün B, Humer S, Moser M (2018) Bivariate jointness measures in Bayesian Model Averaging: Solving the conundrum. Journal of Macroeconomics, 57, 150-165. doi: 10.1016/j.jmacro.2018.05.005
   Ley E, Steel M (2007) Jointness in Bayesian variable selection with applications to growth regression. Journal of Macroeconomics, 29(3), 476-493. doi: 10.1016/j.jmacro.2006.12.002

Usage

```
jointness(bma_list, measure = "HCGHM", rho = 0.5, round = 3)
```

Arguments

`bma_list`	bma object (the result of the bma function)
`measure`	Parameter for choosing the measure of jointness: HCGHM - for Hofmarcher et al. (2018) measure; LS - for Ley & Steel (2007) measure; DW - for Doppelhofer & Weeks (2009) measure; PPI - for posterior probability of including both variables.
`rho`	The parameter "rho" (ρ) to be used in HCGHM jointness measure (default rho = 0.5). Works only if HCGHM measure is chosen (Hofmarcher et al. 2018).
`round`	Parameter indicating the decimal place to which the jointness measures should be rounded (default round = 3).

Value

A table with jointness measures for all the pairs of regressors used in the analysis. The results obtained with the binomial model prior are above the diagonal, while the ones obtained with the binomial-beta prior are below.

Examples

```
library(magrittr)

data_prepared <- badp::economic_growth[, 1:6] %>%
  badp::feature_standardization(
    excluded_cols = c(country, year, gdp)
  ) %>%
  badp::feature_standardization(
    group_by_col  = year,
    excluded_cols = country,
    scale         = FALSE
  )

bma_results <- bma(
  model_space = badp::small_model_space,
  round       = 3,
  dilution    = 0
)

jointness_table <- jointness(bma_results, measure = "HCGHM", rho = 0.5, round = 3)
```

---

join_lagged_col *Dataframe with no lagged column*

Description

This function allows to turn data in the format with lagged values for a chosen column (i.e. there are two columns with the same quantity, but one column is lagged in time) into the format with just one column

Usage

```
join_lagged_col(
  df,
  col,
  col_lagged,
  timestamp_col,
  entity_col,
  timestep = NULL
)
```

Arguments

df	Dataframe with data with a column with lagged values
col	Column with quantity not lagged
col_lagged	Column with the same quantity as col, but the values are lagged in time
timestamp_col	Column with timestamps (e.g. years)
entity_col	Column with entities (e.g. countries)
timestep	Difference between timestamps (e.g. 10)

Value

A dataframe with two columns merged, i.e. just one column with the desired quantity is left.

Examples

```
df <- data.frame(
  year = c(2000, 2001, 2002, 2003, 2004),
  country = c("A", "A", "B", "B", "C"),
  gdp = c(1, 2, 3, 4, 5),
  gdp_lagged = c(NA, 1, 2, 3, 4)
)

join_lagged_col(df, gdp, gdp_lagged, year, country, 1)
```

---

matrices_from_df *List of matrices for SEM model*

Description

List of matrices for SEM model

Usage

```
matrices_from_df(
  df,
  timestamp_col,
  entity_col,
  dep_var_col,
  lin_related_regressors = NULL,
  which_matrices = c("Y1", "Y2", "Z", "cur_Y2", "cur_Z", "res_maker_matrix")
)
```

Arguments

df	Dataframe with data for the likelihood computations.
timestamp_col	Column which determines time stamps. For now only natural numbers can be used.
entity_col	Column which determines entities (e.g. countries, people)
dep_var_col	Column with dependent variable
lin_related_regressors	Vector of strings of column names. Which subset of regressors is in non trivial linear relation with the dependent variable (dep_var_col). In other words regressors with non-zero beta parameters.
which_matrices	character vector with names of matrices which should be computed. Possible matrices are "Y1", "Y2", "Z", "cur_Y2", "cur_Z", "res_maker_matrix". Default is c("Y1", "Y2", "Z", "cur_Y2","cur_Z", "res_maker_matrix") in which case all possible matrices are generated

Value

Named list with matrices as its elements

Examples

```
matrices_from_df(economic_growth, year, country, gdp, c("pop", "sed"),
                 c("Y1", "Y2"))
```

---

model_pmp *Graphs of the prior and posterior model probabilities for the best individual models*

Description

This function draws four graphs of prior and posterior model probabilities for the best individual models:
a) The results with binomial model prior (based on PMP - posterior model probability)
b) The results with binomial-beta model prior (based on PMP - posterior model probability)
Models on the graph are ordered according to their posterior model probability.

Arguments

bma_list	bma_list object (the result of the bma function)
top	The number of the best model to be placed on the graphs

Value

A list with three graphs with prior and posterior model probabilities for individual models:

1. The results with binomial model prior (based on PMP - posterior model probability)
2. The results with binomial-beta model prior (based on PMP - posterior model probability)
3. On graph combining the aforementioned graphs

Examples

```
library(magrittr)

data_prepared <- badp::economic_growth[, 1:6] %>%
  badp::feature_standardization(
    excluded_cols = c(country, year, gdp)
  ) %>%
  badp::feature_standardization(
    group_by_col  = year,
    excluded_cols = country,
    scale         = FALSE
  )

bma_results <- bma(
  model_space = badp::small_model_space,
  round       = 3,
  dilution    = 0
)

model_graphs <- model_pmp(bma_results, top = 16)
```

`model_sizes` *Graphs of the prior and posterior model probabilities of the model sizes*

Description

This function draws four graphs of prior and posterior model probabilities:
a) The results with binomial model prior (based on PMP - posterior model probability)
b) The results with binomial-beta model prior (based on PMP - posterior model probability)

Arguments

`bma_list`	bma_list object (the result of the bma function)

Value

A list with three graphs with prior and posterior model probabilities for model sizes:

1. The results with binomial model prior (based on PMP - posterior model probability)
2. The results with binomial-beta model prior (based on PMP - posterior model probability)
3. One graph combining all the aforementioned graphs

Examples

```
library(magrittr)

data_prepared <- badp::economic_growth[, 1:6] %>%
  badp::feature_standardization(
    excluded_cols = c(country, year, gdp)
  ) %>%
  badp::feature_standardization(
    group_by_col  = year,
    excluded_cols = country,
    scale         = FALSE
  )

bma_results <- bma(
  model_space = badp::small_model_space,
  round       = 3,
  dilution    = 0
)

size_graphs <- model_sizes(bma_results)
```

---

model_space_nonnested *Example output of* optim_model_space *for non-nested models*

Description

A list created with optim_model_space using the economic_growth dataset.

Usage

```
model_space_nonnested
```

Format

A list with 5 elements:

params A numeric matrix with 40 rows and 512 columns, containing parameter values for the model space. Each column represents a different model.

stats A numeric matrix of statistics computed by `compute_model_space_stats` based on `params`. Row 1 contains model likelihoods. Row 2 contains a quantity proportional to `0.5 * BIC` (cf. Raftery, Bayesian Model Selection in Social Research, Eq. 19). Rows 3–7 contain standard deviations, and rows 8–12 contain robust standard deviations.

reg_names A character vector with the names of the variables.

observations_num The total number of observations in the panel (292).

df The data frame used in the analysis.

`nested_optimization_wrapper`

Helper-function - finds parameters minimizing log-likelihood function for the nested version of the SEM setup, using BFGS method

---

Description

Helper-function - finds parameters minimizing log-likelihood function for the nested version of the SEM setup, using BFGS method

Usage

```
nested_optimization_wrapper(
  params,
  df,
  timestamp_col,
  entity_col,
  dep_var_col,
  data,
  exact_value,
  control
)
```

Arguments

`params`	Vector of the initial parameters
`df`	Data frame with data for the SEM analysis.
`timestamp_col`	Column which determines time periods. For now only natural numbers can be used as timestamps
`entity_col`	Column which determines entities (e.g. countries, people)
`dep_var_col`	Column with dependent variable

data	List of SEM setup matrices shared along the models
exact_value	Whether the exact value of the likelihood should be computed (TRUE) or just the proportional part (FALSE). Check sem_likelihood for details.
control	a list of control parameters for the optimization which are passed to optim. Default is list(trace = 2, maxit = 10000, fnscale = -1, REPORT = 100, scale = 0.05).

Value

List (or matrix) of parameters describing analyzed models.

---

nested_std_dev_from_params

Helper function - wraps single execution of the log-likelihood & deviation parameters calculations. Used for non nested version of SEM likelihood.

Description

Helper function - wraps single execution of the log-likelihood & deviation parameters calculations. Used for non nested version of SEM likelihood.

Usage

```
nested_std_dev_from_params(
  params,
  data,
  df,
  timestamp_col,
  entity_col,
  dep_var_col,
  n_entities,
  periods_n
)
```

Arguments

params	A matrix (with named rows) with each column corresponding to a model. Each row specifies model parameters. Compare with optim_model_space_params
data	List of the SEM setup matrices, shared along different models
df	Data frame with data for the SEM analysis.
timestamp_col	The name of the column with timestamps
entity_col	Column with entities (e.g. countries)
dep_var_col	Column with the dependent variable
n_entities	Number of entities - passed to save calc. time
periods_n	Number of periods - passed to save calc. time

Value

#' Matrix with columns describing likelihood and standard deviations for each model. The first row is the likelihood for the model (computed using the parameters in the provided model space). The second row is almost 1/2 * BIC_k as in Raftery's Bayesian Model Selection in Social Research eq. 19 (see TODO in the code below). The third row is model posterior probability. Then there are rows with standard deviations for each parameter. After that we have rows with robust standard deviation (not sure yet what exactly "robust" means).

`non_nested_optimization_wrapper`

Helper-function - finds parameters minimizing log-likelihood function for the non-nested version of the SEM setup, using BFGS method

---

Description

Helper-function - finds parameters minimizing log-likelihood function for the non-nested version of the SEM setup, using BFGS method

Usage

```
non_nested_optimization_wrapper(
  params,
  df,
  timestamp_col,
  entity_col,
  dep_var_col,
  exact_value,
  n_all_regressors,
  n_timestamp,
  control
)
```

Arguments

`params`	Vector of the initial parameters
`df`	Data frame with data for the SEM analysis.
`timestamp_col`	Column which determines time periods. For now only natural numbers can be used as timestamps
`entity_col`	Column which determines entities (e.g. countries, people)
`dep_var_col`	Column with dependent variable
`exact_value`	Whether the exact value of the likelihood should be computed (`TRUE`) or just the proportional part (`FALSE`). Check sem_likelihood for details.

`n_all_regressors`
: Integer. Total number of potential regressors in the full (maximal) model space. Used to compute the full parameter dimension (for ϕ and ψ) so that parameters corresponding to excluded regressors can be padded with `NA` in the non-nested setup.

`n_timestamp`
: Integer. Number of time periods in the panel (i.e. the number of distinct values in `timestamp_col`). Used to determine the required number of ϕ and ψ parameters for the current model and for the full model.

`control`
: a list of control parameters for the optimization which are passed to optim. Default is `list(trace = 2, maxit = 10000, fnscale = -1, REPORT = 100, scale = 0.05)`.

Value

List (or matrix) of parameters describing analyzed models.

`non_nested_std_dev_from_params`
: *Helper function - wraps single execution of the log-likelihood & deviation parameters calculations. Used for non nested version of SEM likelihood.*

Description

Helper function - wraps single execution of the log-likelihood & deviation parameters calculations. Used for non nested version of SEM likelihood.

Usage

```
non_nested_std_dev_from_params(
  params,
  df,
  timestamp_col,
  entity_col,
  dep_var_col,
  n_entities,
  periods_n
)
```

Arguments

`params`
: A matrix (with named rows) with each column corresponding to a model. Each row specifies model parameters. Compare with optim_model_space_params

`df`
: Data frame with data for the SEM analysis.

`timestamp_col`
: The name of the column with timestamps

`entity_col`
: Column with entities (e.g. countries)

dep_var_col	Column with the dependent variable
n_entities	Number of entities - passed to save calc. time
periods_n	Number of periods - passed to save calc. time

Value

#' Matrix with columns describing likelihood and standard deviations for each model. The first row is the likelihood for the model (computed using the parameters in the provided model space). The second row is almost 1/2 * BIC_k as in Raftery's Bayesian Model Selection in Social Research eq. 19 (see TODO in the code below). The third row is model posterior probability. Then there are rows with standard deviations for each parameter. After that we have rows with robust standard deviation (not sure yet what exactly "robust" means).

---

optim_model_space *Calculation of the model_space object*

Description

This function calculates model space, values of the maximized likelihood function, BICs, and standard deviations of the parameters that will be used in Bayesian model averaging. Moreover, it provides a vector with the names of the variables for bma function and the number of observations.

Usage

```
optim_model_space(
  df,
  timestamp_col,
  entity_col,
  dep_var_col,
  init_value,
  nested = TRUE,
  exact_value = FALSE,
  cl = NULL,
 control = list(trace = 0, maxit = 10000, fnscale = -1, REPORT = 100, scale = 0.05)
)
```

Arguments

df	Data frame with data for the analysis.
timestamp_col	The name of the column with time stamps
entity_col	Column with entities (e.g. countries)
dep_var_col	Column with the dependent variable
init_value	The value with which the model space will be initialized. This will be the starting point for the numerical optimization.

nested	Logical. If TRUE (default), compute approximate standard deviations using the nested-model approach via nested_std_dev_from_params(). If FALSE, use the non-nested approach via non_nested_std_dev_from_params(). The choice affects which approximation routine is used for each model in params.
exact_value	Whether the exact value of the likelihood should be computed (TRUE) or just the proportional part (FALSE). Check sem_likelihood for details.
cl	An optional cluster object. If supplied, the function will use this cluster for parallel processing. If NULL (the default), pbapply::pblapply will run sequentially.
control	a list of control parameters for the optimization which are passed to optim. Default is list(trace = 2, maxit = 10000, fnscale = -1, REPORT = 100, scale = 0.05), but note that scale is used only for adjusting the parscale element added later in the function code.

Value

List with two objects:

1. params - table with parameters of all estimated models
2. stats - table with the value of maximized likelihood function, BIC, and standard errors for all estimated models
3. reg_names - vector with the names of the variables
4. observations_num - number of observations
5. df - data frame used in estimation

Examples

```
library(magrittr)

data_prepared <- badp::economic_growth[, 1:5] %>%
  badp::feature_standardization(
    excluded_cols = c(country, year, gdp)
  ) %>%
  badp::feature_standardization(
    group_by_col  = year,
    excluded_cols = country,
    scale         = FALSE
  )

optim_model_space(
  df            = data_prepared,
  dep_var_col   = gdp,
  timestamp_col = year,
  entity_col    = country,
```

```
  init_value    = 0.5
)
```

`optim_model_space_params` *Finds MLE parameters for each model in the given model space*

Description

Given a dataset and an initial value for parameters, initializes a model space with parameters equal to the initial value for each model. Then for each model performs a numerical optimization and finds parameters which maximize the likelihood.

Usage

```
optim_model_space_params(
  df,
  timestamp_col,
  entity_col,
  dep_var_col,
  init_value,
  nested,
  exact_value = FALSE,
  cl = NULL,
 control = list(trace = 0, maxit = 10000, fnscale = -1, REPORT = 100, scale = 0.05)
)
```

Arguments

`df`	Data frame with data for the analysis.
`timestamp_col`	The name of the column with time stamps.
`entity_col`	Column with entities (e.g. countries).
`dep_var_col`	Column with the dependent variable.
`init_value`	The value with which the model space will be initialized. This will be the starting point for the numerical optimization.
`nested`	Logical. If `TRUE` (default), compute approximate standard deviations using the nested-model approach via `nested_std_dev_from_params()`. If `FALSE`, use the non-nested approach via `non_nested_std_dev_from_params()`. The choice affects which approximation routine is used for each model in `params`.
`exact_value`	Whether the exact value of the likelihood should be computed (`TRUE`) or just the proportional part (`FALSE`). Check sem_likelihood for details.
`cl`	An optional cluster object. If supplied, the function will use this cluster for parallel processing. If `NULL` (the default), `pbapply::pblapply` will run sequentially.

`control` a list of control parameters for the optimization which are passed to optim. Default is `list(trace = 2, maxit = 10000, fnscale = -1, REPORT = 100, scale = 0.05)`.

Value

List (or matrix) of parameters describing analyzed models.

`original_economic_growth`

Economic Growth Data in the original format

Description

Data used in Growth Empirics in Panel Data under Model Uncertainty and Weak Exogeneity (Moral-Benito, 2016, Journal of Applied Econometrics).

Usage

```
original_economic_growth
```

Format

`original_economic_growth`**:**

A data frame with 292 rows and 13 columns (73 countries and 4 periods + extra one for lagged dependent variable):

year Year

country Country ID

gdp Logarithm of GDP per capita (2000 US dollars at PP)

gdp_lag Lagged logarithm of GDP per capita (2000 US dollars at PP)

ish Ratio of real domestic investment to GDP

sed Stock of years of secondary education in the total population

pgrw Average growth rate of population

pop Population in millions of people

ipr Purchasing-power-parity numbers for investment goods

opem Exports plus imports as a share of GDP

gsh Ratio of government consumption to GDP

lnlex Logarithm of the life expectancy at birth

polity Composite index given by the democracy score minus the autocracy score

Source

http://qed.econ.queensu.ca/jae/datasets/moral-benito001/

posterior_dens *Graphs of the posterior densities of the coefficients*

Description

This function draws graphs of the posterior densities of all the coeffcients of interest.

Arguments

bma_list	bma object (the result of the bma function)
prior	Parameter indicating which model prior should be used for calculations: 1. "binomial" - using binomial model prior (default option) 2. "beta" - using binomial-beta model prior
SE	Parameter indicating which standard errors should be used in calculation of posterior standard deviation: 1. "standard" - regular standard errors (default option) 2. "robust" - robust standard errors

Value

A list with the graphs of the posterior densities of coefficients for all the considered regressors.

Examples

```
library(magrittr)

data_prepared <- badp::economic_growth[, 1:6] %>%
  badp::feature_standardization(
    excluded_cols = c(country, year, gdp)
  ) %>%
  badp::feature_standardization(
    group_by_col  = year,
    excluded_cols = country,
    scale         = FALSE
  )

bma_results <- bma(
  model_space = badp::small_model_space,
  round       = 3,
  dilution    = 0
)

posterior_graphs <- posterior_dens(bma_results, prior = "binomial", SE = "robust")
```

regressor_names_from_params_vector *Helper function to extract names from a vector defining a model*

Description

For now it is assumed that we can only exclude linear relationships between regressors and the dependent variable.

Usage

```
regressor_names_from_params_vector(params)
```

Arguments

params	a vector with parameters describing the model

Details

The vector needs to have named rows, i.e. it is assumed it comes from a model space (see init_model_space_params for details).

Value

Names of regressors which are assumed to be linearly connected with dependent variable within the model described by the params vector.

Examples

```
params <- c(alpha = 1, beta_gdp = 1, beta_gdp_lagged = 1, phi_0 = 1, err_var = 1)
regressor_names_from_params_vector(params)
```

residual_maker_matrix *Residual Maker Matrix*

Description

Create residual maker matrix from a given matrix m. See article about projection matrix on the Wikipedia.

Usage

```
residual_maker_matrix(m)
```

Arguments

m	Matrix

Value

M x M matrix where M is the number of rows in the m matrix.

Examples

```
residual_maker_matrix(matrix(c(1,2,3,4), nrow = 2))
```

---

sem_B_matrix *Coefficients matrix for SEM representation*

Description

Create coefficients matrix for Simultaneous Equations Model (SEM) representation.

Usage

```
sem_B_matrix(alpha, periods_n, beta = NULL)
```

Arguments

alpha	numeric
periods_n	integer
beta	numeric vector. Default is c() for no regressors case.

Value

List with two matrices B11 and B12

Examples

```
sem_B_matrix(3, 4, 4:6)
```

sem_C_matrix *Coefficients matrix for initial conditions*

Description

Create matrix for Simultaneous Equations Model (SEM) representation with coefficients placed next to initial values of regressors, dependent variable and country-specific time-invariant variables.

Usage

```
sem_C_matrix(alpha, phi_0, periods_n, beta = NULL, phi_1 = NULL)
```

Arguments

alpha	numeric
phi_0	numeric
periods_n	numeric
beta	numeric vector. Default is c() for no regressors case.
phi_1	numeric vector. Default is c() for no regressors case.

Value

matrix

Examples

```
alpha <- 9
phi_0 <- 19
beta <- 11:15
phi_1 <- 21:25
periods_n <- 4
sem_C_matrix(alpha, phi_0, periods_n, beta, phi_1)
```

sem_dep_var_matrix *Matrix with dependent variable data for SEM representation*

Description

Create matrix which contains dependent variable data used in the Simultaneous Equations Model (SEM) representation on the left hand side of the equations. The matrix contains the data for time periods greater than or equal to the second lowest time stamp. The matrix is then used to compute likelihood for SEM analysis.

Usage

```
sem_dep_var_matrix(df, timestamp_col, entity_col, dep_var_col)
```

Arguments

df	Data frame with data for the SEM analysis.
timestamp_col	Column which determines time periods. For now only natural numbers can be used as timestamps
entity_col	Column which determines entities (e.g. countries, people)
dep_var_col	Column with dependent variable

Value

Matrix of size N x T where N is the number of entities considered and T is the number of periods greater than or equal to the second lowest time stamp.

Examples

```
set.seed(1)
df <- data.frame(
  entities = rep(1:4, 5),
  times = rep(seq(1960, 2000, 10), each = 4),
  dep_var = stats::rnorm(20), a = stats::rnorm(20), b = stats::rnorm(20)
)
sem_dep_var_matrix(df, times, entities, dep_var)
```

---

sem_likelihood *Likelihood for the SEM model*

Description

Likelihood for the SEM model

Usage

```
sem_likelihood(
  params,
  data,
  timestamp_col,
  entity_col,
  dep_var_col,
  lin_related_regressors = NULL,
  per_entity = FALSE,
  exact_value = TRUE
)
```

Arguments

params	Parameters describing the model. Can be either a vector or a list with named parameters. See 'Details'
data	Data for the likelihood computations. Can be either a list of matrices or a dataframe. If the dataframe, additional parameters are required to build the matrices within the function.
timestamp_col	Column which determines time stamps. For now only natural numbers can be used.
entity_col	Column which determines entities (e.g. countries, people)
dep_var_col	Column with dependent variable
lin_related_regressors	Which subset of columns should be used as regressors for the current model. In other words regressors are the total set of regressors and lin_related_regressors are the ones for which linear relation is not set to zero for a given model.
per_entity	Whether to compute overall likelihood or a vector of likelihoods with per entity value
exact_value	Whether the exact value of the likelihood should be computed (TRUE) or just the proportional part (FALSE). Currently TRUE adds: 1. a normalization constant coming from Gaussian distribution, 2. a term disappearing during likelihood simplification in Likelihood-based Estimation of Dynamic Panels with Predetermined Regressors by Moral-Benito (see Appendix A.1). The latter happens when transitioning from equation (47) to equation (48), in step 2: the term trace(HG_22) is dropped, because it can be assumed to be constant from Moral-Benito perspective. To get the exact value of the likelihood we have to take this term into account.

Details

The params argument is a list that should contain the following components:

alpha scalar value which determines linear dependence on lagged dependent variable

phi_0 scalar value which determines linear dependence on the value of dependent variable at the lowest time stamp

err_var scalar value which determines classical error component (Sigma11 matrix, sigma_epsilon^2)

dep_vars double vector of length equal to the number of time stamps (i.e. time stamps greater than or equal to the second lowest time stamp)

beta double vector which determines the linear dependence on regressors different than the lagged dependent variable; The vector should have length equal to the number of regressors.

phi_1 double vector which determines the linear dependence on initial values of regressors different than the lagged dependent variable; The vector should have length equal to the number of regressors.

phis double vector which together with psis determines upper right and bottom left part of the covariance matrix; The vector should have length equal to the number of regressors times number of time stamps minus 1, i.e. regressors_n * (periods_n - 1)

psis double vector which together with psis determines upper right and bottom left part of the covariance matrix; The vector should have length equal to the number of regressors times number of

time stamps minus 1 times number of time stamps divided by 2, i.e. `regressors_n * (periods_n - 1) * periods_n / 2`

Value

The value of the likelihood for SEM model (or a part of interest of the likelihood)

Examples

```
set.seed(1)
df <- data.frame(
  entities = rep(1:4, 5),
  times = rep(seq(1960, 2000, 10), each = 4),
  dep_var = stats::rnorm(20), a = stats::rnorm(20), b = stats::rnorm(20)
)
df <-
  feature_standardization(df, excluded_cols = c(times, entities))
sem_likelihood(0.5, df, times, entities, dep_var)
```

---

sem_psi_matrix *Matrix with psi parameters for SEM representation*

Description

Matrix with psi parameters for SEM representation

Usage

```
sem_psi_matrix(psis, timestamps_n, features_n)
```

Arguments

`psis`	double vector with psi parameter values
`timestamps_n`	number of time stamps (e.g. years)
`features_n`	number of features (e.g. population size, investment rate)

Value

A matrix with `timestamps_n` rows and `(timestamps_n - 1) * feature_n` columns. Psis are filled in row by row in a block manner, i.e. blocks of size `feature_n` are placed next to each other

Examples

```
sem_psi_matrix(1:30, 4, 5)
```

`sem_regressors_matrix` *Matrix with regressors data for SEM representation*

Description

Create matrix which contains regressors data used in the Simultaneous Equations Model (SEM) representation on the left hand side of the equations. The matrix contains regressors data for time periods greater than or equal to the second lowest time stamp. The matrix is then used to compute likelihood for SEM analysis.

Usage

```
sem_regressors_matrix(df, timestamp_col, entity_col, dep_var_col)
```

Arguments

`df`	Data frame with data for the SEM analysis.
`timestamp_col`	Column which determines time periods. For now only natural numbers can be used as timestamps
`entity_col`	Column which determines entities (e.g. countries, people)
`dep_var_col`	Column with dependent variable

Value

Matrix of size N x (T-1)*k where N is the number of entities considered, T is the number of periods greater than or equal to the second lowest time stamp and k is the number of chosen regressors. If there are no regressors returns `NULL`.

Examples

```
set.seed(1)
df <- data.frame(
  entities = rep(1:4, 5),
  times = rep(seq(1960, 2000, 10), each = 4),
  dep_var = stats::rnorm(20), a = stats::rnorm(20), b = stats::rnorm(20)
)
sem_regressors_matrix(df, times, entities, dep_var)
```

sem_sigma_matrix *Covariance matrix for SEM representation*

Description

Create covariance matrix for Simultaneous Equations Model (SEM) representation. Only the part necessary to compute concentrated likelihood function is computed (cf. Appendix in the Moral-Benito paper)

Usage

```
sem_sigma_matrix(err_var, dep_vars, phis = NULL, psis = NULL)
```

Arguments

err_var	numeric
dep_vars	numeric vector
phis	numeric vector
psis	numeric vector

Value

List with two matrices Sigma11 and Sigma12

Examples

```
err_var <- 1
dep_vars <- c(2, 2, 2, 2)
phis <- c(10, 10, 20, 20, 30, 30)
psis <- c(101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112)
sem_sigma_matrix(err_var, dep_vars, phis, psis)
```

small_model_space *Example output of* optim_model_space *(small version)*

Description

A list created with optim_model_space using the economic_growth dataset and only three regressors: ish, sed, and pgrw.

Usage

```
small_model_space
```

Format

A list with 5 elements:

params A numeric matrix with 40 rows and 8 columns (corresponding to $2^3 = 8$ models), containing parameter values for the model space. Each column represents a different model.

stats A numeric matrix of statistics computed by `compute_model_space_stats` based on `params`. Row 1 contains model likelihoods. Row 2 contains a quantity proportional to `0.5 * BIC` (cf. Raftery, Bayesian Model Selection in Social Research, Eq. 19). Rows 3–7 contain standard deviations, and rows 8–12 contain robust standard deviations.

reg_names A character vector with the names of the regressors.

observations_num The total number of observations in the panel (292).

df The data frame used in the analysis.

Index