Package 'lpme'

January 25, 2025

Title Measurement Error Analysis and Correction Under Identification Restrictions

Version 0.1

Description An R package for analyzing latent variable models with measurement error correction, including Item Response Theory (IRT) models. It provides tools for implementing various correction methods such as Bayesian MCMC, overimputation, bootstrapping for robust standard errors, OLS, and IV-based approaches. The package supports flexible specification of observable indicators and groupings, making it suitable for latent variable analyses in social sciences and other fields.

Depends R (>= 3.3.3)

License GLP3

Encoding UTF-8

LazyData false

Maintainer Connor Jerzak <connor.jerzak@gmail.com>

Imports reticulate,
stats,
sensemakr,
pscl,
AER,
sandwich,
mvtnorm,
Amelia,
emIRT,
gtools

Suggests testthat (>= 3.0.0),
knitr,
rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.2

URL https://github.com/cjerzak/lpme

BugReports https://github.com/cjerzak/lpme/issues

Contents

`build_backend` *A function to build the environment for lpme. Builds a conda environment in which 'JAX' and 'np' are installed. Users can also create a conda environment where 'JAX' and 'np' are installed themselves.*

Description

A function to build the environment for lpme. Builds a conda environment in which 'JAX' and 'np' are installed. Users can also create a conda environment where 'JAX' and 'np' are installed themselves.

Usage

```
build_backend(conda_env = "lpme", conda = "auto")
```

Arguments

`conda_env`	(default = `"lpme"`) Name of the conda environment in which to place the backends.
`conda`	(default = `auto`) The path to a conda executable. Using `"auto"` allows reticulate to attempt to automatically find an appropriate conda binary.

Value

Invisibly returns NULL; this function is used for its side effects of creating and configuring a conda environment for `lpme`. This function requires an Internet connection. You can find out a list of conda Python paths via: `Sys.which("python")`

Examples

```
## Not run:
# Create a conda environment named "lpme"
# and install the required Python packages (jax, numpy, etc.)
build_backend(conda_env = "lpme", conda = "auto")

# If you want to specify a particular conda path:
# build_backend(conda_env = "lpme", conda = "/usr/local/bin/conda")

## End(Not run)
```

lpme *lpme*

Description

Implements bootstrapped analysis for latent variable models with measurement error correction

Usage

```
lpme(
  Y,
  observables,
  observables_groupings = colnames(observables),
  make_observables_groupings = FALSE,
  n_boot = 32L,
  n_partition = 10L,
  boot_basis = 1:length(Y),
  return_intermediaries = TRUE,
  estimation_method = "emIRT",
  ordinal = FALSE,
  conda_env = "lpme",
  conda_env_required = TRUE
)
```

Arguments

`Y` A vector of observed outcome variables

`observables` A matrix of observable indicators used to estimate the latent variable

`observables_groupings`
A vector specifying groupings for the observable indicators. Default is column names of observables.

`make_observables_groupings`
Logical. If TRUE, creates dummy variables for each level of the observable indicators. Default is FALSE.

`n_boot` Integer. Number of bootstrap iterations. Default is 32.

`n_partition` Integer. Number of partitions for each bootstrap iteration. Default is 10.

`boot_basis` Vector of indices or grouping variable for stratified bootstrap. Default is 1:length(Y).

`return_intermediaries`
Logical. If TRUE, returns intermediate results. Default is TRUE.

`estimation_method`
Character specifying the estimation approach. Options include:

- "emIRT" (default): Uses expectation-maximization via `emIRT` package. Supports both binary (via `emIRT::binIRT`) and ordinal (via `emIRT::ordIRT`) indicators.
- "MCMC": Markov Chain Monte Carlo estimation using either `pscl::ideal` (R backend) or `numpyro` (Python backend)
- "MCMCFull": Full Bayesian model that simultaneously estimates latent variables and outcome relationship using `numpyro`

- "MCMCOverImputation": Two-stage MCMC approach with measurement error correction via over-imputation

ordinal
: Logical indicating whether the observable indicators are ordinal (TRUE) or binary (FALSE).

conda_env
: A character string specifying the name of the conda environment to use via reticulate. Default is "lpme".

conda_env_required
: A logical indicating whether the specified conda environment must be strictly used. If TRUE, an error is thrown if the environment is not found. Default is TRUE.

Details

This function implements a bootstrapped latent variable analysis with measurement error correction. It performs multiple bootstrap iterations, each with multiple partitions. For each partition, it calls the LatentOneRun function to estimate latent variables and apply various correction methods. The results are then aggregated across partitions and bootstrap iterations to produce final estimates and bootstrap standard errors.

Value

A list containing various estimates and statistics (in snake_case):

- `ols_coef`: Coefficient from naive OLS regression.
- `ols_se`: Standard error of naive OLS coefficient.
- `ols_tstat`: T-statistic of naive OLS coefficient.
- `iv_coef`: Coefficient from instrumental variable (IV) regression.
- `iv_se`: Standard error of IV regression coefficient.
- `iv_tstat`: T-statistic of IV regression coefficient.
- `corrected_iv_coef`: IV regression coefficient corrected for measurement error.
- `corrected_iv_se`: Standard error of the corrected IV coefficient (currently `NA`).
- `corrected_iv_tstat`: T-statistic of the corrected IV coefficient.
- `var_est`: Estimated variance of the measurement error (split-half variance).
- `corrected_ols_coef`: OLS coefficient corrected for measurement error.
- `corrected_ols_se`: Standard error of the corrected OLS coefficient (currently `NA`).
- `corrected_ols_tstat`: T-statistic of the corrected OLS coefficient (currently `NA`).
- `corrected_ols_coef_alt`: Alternative corrected OLS coefficient (if applicable).
- `corrected_ols_se_alt`: Standard error for the alternative corrected OLS coefficient (if applicable).
- `corrected_ols_tstat_alt`: T-statistic for the alternative corrected OLS coefficient (if applicable).
- `bayesian_ols_coef_outer_normed`: Posterior mean of the OLS coefficient under MCMC, after normalizing by the overall sample standard deviation.
- `bayesian_ols_se_outer_normed`: Posterior standard error corresponding to `bayesian_ols_coef_outer_normed`
- `bayesian_ols_tstat_outer_normed`: T-statistic for `bayesian_ols_coef_outer_normed`.
- `bayesian_ols_coef_inner_normed`: Posterior mean of the OLS coefficient under MCMC, after normalizing each posterior draw individually.

- bayesian_ols_se_inner_normed: Posterior standard error corresponding to bayesian_ols_coef_inner_normed
- bayesian_ols_tstat_inner_normed: T-statistic for bayesian_ols_coef_inner_normed.
- m_stage_1_erv: Extreme robustness value (ERV) for the first-stage regression (x_est2 on x_est1), if computed.
- m_reduced_erv: ERV for the reduced model (Y on x_est1), if computed.
- x_est1: First set of latent variable estimates.
- x_est2: Second set of latent variable estimates.

Examples

```
# Generate some example data
set.seed(123)
Y <- rnorm(1000)
observables <- as.data.frame( matrix(sample(c(0,1), 1000*10, replace = TRUE), ncol = 10) )

# Run the bootstrapped analysis
results <- lpme(Y, observables, n_boot = 100, n_partition = 5)

# View the corrected IV coefficient and its standard error
print(c(results$corrected_iv_reg_coef, results$corrected_iv_reg_se))
```

---

lpme_onerun *lpme_onerun*

Description

Implements analysis for latent variable models with measurement error correction

Usage

```
lpme_onerun(
  Y,
  observables,
  observables_groupings = colnames(observables),
  make_observables_groupings = FALSE,
  estimation_method = "emIRT",
  ordinal = FALSE,
  conda_env = "lpme",
  conda_env_required = TRUE,
  seed = NULL
)
```

Arguments

Y	A vector of observed outcome variables
observables	A matrix of observable indicators used to estimate the latent variable
observables_groupings	A vector specifying groupings for the observable indicators. Default is column names of observables.

make_observables_groupings
: Logical. If TRUE, creates dummy variables for each level of the observable indicators. Default is FALSE.

estimation_method
: Character specifying the estimation approach. Options include:
  - "emIRT" (default): Uses expectation-maximization via emIRT package. Supports both binary (via emIRT::binIRT) and ordinal (via emIRT::ordIRT) indicators.
  - "MCMC": Basic Markov Chain Monte Carlo estimation using either pscl::ideal (R backend) or numpyro (Python backend)
  - "MCMCFull": Full Bayesian model that simultaneously estimates latent variables and outcome relationship using numpyro
  - "MCMCOverImputation": Two-stage MCMC approach with measurement error correction via over-imputation

ordinal
: Logical indicating whether the observable indicators are ordinal (TRUE) or binary (FALSE).

conda_env
: A character string specifying the name of the conda environment to use via reticulate. Default is "lpme".

conda_env_required
: A logical indicating whether the specified conda environment must be strictly used. If TRUE, an error is thrown if the environment is not found. Default is TRUE.

seed
: Random seed for reproducibility. Default is a random integer between 1 and 10000 (used internally)

Details

This function implements a latent variable analysis with measurement error correction. It splits the observable indicators into two sets, estimates latent variables using each set, and then applies various correction methods including OLS correction and instrumental variable approaches.

Value

A list containing various estimates and statistics:

- ols_coef: Coefficient from naive OLS regression
- ols_se: Standard error of naive OLS coefficient
- ols_tstat: T-statistic of naive OLS coefficient
- corrected_ols_coef: OLS coefficient corrected for measurement error
- corrected_ols_se: Standard error of corrected OLS coefficient (currently NA)
- corrected_ols_tstat: T-statistic of corrected OLS coefficient (currently NA)
- corrected_ols_coef_alt: Alternative corrected OLS coefficient (currently NA)
- iv_coef: Coefficient from instrumental variable regression
- iv_se: Standard error of IV regression coefficient
- iv_tstat: T-statistic of IV regression coefficient
- corrected_iv_coef: IV regression coefficient corrected for measurement error
- corrected_iv_se: Standard error of corrected IV coefficient
- corrected_iv_tstat: T-statistic of corrected IV coefficient

- `var_est_split`: Estimated variance of the measurement error
- `x_est1`: First set of latent variable estimates
- `x_est2`: Second set of latent variable estimates

Examples

```
# Generate some example data
set.seed(123)
Y <- rnorm(100)
observables <- as.data.frame( matrix(sample(c(0,1), 1000*10, replacement = TRUE), ncol = 10) )

# Run the analysis
results <- lpme_onerun(Y, observables)

# View the corrected OLS coefficient
print(results$Corrected_OLSCoef)
```

Index