Package 'MultiLevelOptimalBayes'

October 11, 2026

Type Package

Title Regularized Bayesian Estimator for Two-Level Latent Variable Models

Version 0.1.0.0

Author Valerii Dashuk [aut, cre],
Binayak Timilsina [aut],
Martin Hecht [aut],
Steffen Zitzmann [aut]

Maintainer Valerii Dashuk <vadashuk@gmail.com>

Description Implements a regularized Bayesian estimator that optimizes the estimation of between-group coefficients for multilevel latent variable models by minimizing mean squared error (MSE) and balancing variance and bias. The package provides more reliable estimates in scenarios with limited data, offering a robust solution for accurate parameter estimation in two-level latent variable models. It is designed for researchers in psychology, education, and related fields who face challenges in estimating between-group effects under small sample sizes and low intraclass correlation coefficients. The package includes comprehensive S3 methods for result objects: print(), summary(), coef(), se(), vcov(), confint(), as.data.frame(), dim(), length(), names(), and update() for enhanced usability and integration with standard R workflows. Dashuk et al. (2025a) <doi:10.1017/psy.2025.10045> derived the optimal regularized Bayesian estimator; Dashuk et al. (2025b) <doi:10.1007/s41237-025-00264-7> extended it to the multivariate case; and Luedtke et al. (2008) <doi:10.1037/a0012869> formalized the two-level latent variable framework.

License GPL-3

Depends R (>= 4.1.0)

Encoding UTF-8

RoxygenNote 7.3.2

Suggests knitr, lme4, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2026-10-11 13:50:16 UTC

Contents

as.data.frame.mlob_result 2
coef.mlob_result 3
confint.mlob_result 3
dim.mlob_result 4
length.mlob_result 4
mlob 5
names.mlob_result 8
se 9
se.mlob_result 9
summary.mlob_result 10
update.mlob_result 10
vcov.mlob_result 11

Index **12**

`as.data.frame.mlob_result` *Convert mlob_result to data frame*

Description

Convert mlob_result to data frame

Usage

```
## S3 method for class 'mlob_result'
as.data.frame(x, ...)
```

Arguments

x	An object of class mlob_result
...	Additional arguments passed to as.data.frame

Value

A data frame representation of the results

coef.mlob_result *Extract coefficients from mlob_result objects*

Description

Extract coefficients from mlob_result objects

Usage

```
## S3 method for class 'mlob_result'
coef(object, ...)
```

Arguments

object	An object of class mlob_result
...	Additional arguments passed to coef

Value

A data frame with coefficients

confint.mlob_result *Extract confidence intervals from mlob_result objects*

Description

Extract confidence intervals from mlob_result objects

Usage

```
## S3 method for class 'mlob_result'
confint(object, parm, level = 0.95, ...)
```

Arguments

object	An object of class mlob_result
parm	Parameters for which to extract confidence intervals
level	Confidence level (can be different from the model's default)
...	Additional arguments passed to confint

Value

A matrix with confidence intervals

dim.mlob_result *Get dimensions of mlob_result objects*

Description

Get dimensions of mlob_result objects

Usage

```
## S3 method for class 'mlob_result'
dim(x)
```

Arguments

x	An object of class mlob_result

Value

A vector with dimensions

length.mlob_result *Get length of mlob_result objects*

Description

Get length of mlob_result objects

Usage

```
## S3 method for class 'mlob_result'
length(x)
```

Arguments

x	An object of class mlob_result

Value

The number of parameters/coefficients

mlob *MultiLevel Optimal Bayes Function (MLOB)*

Description

Implements a regularized Bayesian approach that optimizes the estimation of between-group coefficients by minimizing Mean Squared Error (MSE), balancing both variance and bias. This method provides more reliable estimates in scenarios with limited data, offering a robust solution for accurate parameter estimation in multilevel models. The package is designed for researchers in psychology, education, and related fields who face challenges in estimating between-group effects in two-level latent variable models, particularly in scenarios with small sample sizes and low intraclass correlation coefficients.

Usage

```
mlob(
  formula,
  data,
  group,
  balancing.limit = 0.2,
  conf.level = 0.95,
  jackknife = FALSE,
  punish.coeff = 2,
  ...
)
```

Arguments

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. Formula specifies the model (e.g., `Y ~ X + C...`), where Y is the dependent variable, X is the context variable, which is the focus of most applications of the model (always included), and C includes all additional covariates.
`data`	a data frame (or object converted by as.data.frame to a data frame) containing the variables referenced in the formula. All variables used in the model, including the dependent variable, context variable, covariates, and grouping variable must be present in this data frame.
`group`	a name of the variable that defines the affiliation of an individual (row) to the specific group.
`balancing.limit`	a number that represents the threshold of the maximum relative part of the dataset that can be deleted to balance the data. Defaults to `0.2`
`conf.level`	a numeric value representing the confidence level used to calculate confidence intervals for the estimators. Defaults to `0.95`, corresponding to a `95%` confidence level.

jackknife	logical variable. If TRUE, the jackknife re-sampling method will be applied to calculate the standard error of the between-group and its confidence interval coefficient. Defaults to FALSE.
punish.coeff	a multiplier that punishes the balancing procedure when deleting the whole group. If punish.coeff is equal to 1, no additional punishment is applied for deleting the group. Higher values intensify the penalty. Defaults to 2.
...	additional arguments passed to the function.

Details

This function also verifies whether the data is balanced (i.e., whether each group contains the same number of individuals). If the data is unbalanced, the balancing procedure comes into effect, and identifies the optimal number of items and groups to delete based on the punishment coefficient. If the amount of data deleted is more than defined by threshold (balancing.limit) then results should be interpreted with caution.

The summary() function produces output similar to:

```
Summary of Coefficients:
                    Estimate Std. Error Lower CI (99
beta_b         0.2957937 0.30457475    -0.4887389     1.080326 0.9711694 0.331464
gamma_Petal.Length 0.4679522 0.05038678    0.3381645     0.597740 9.2872029 0.000000       ***

For comparison, summary of coefficients from unoptimized analysis (ML):
                    Estimate Std. Error Lower CI (99
beta_b         0.6027440 0.87389866    -1.6482698     2.853758 0.6897184 0.4903713
gamma_Petal.Length 0.4679522 0.05038678    0.3381645     0.597740 9.2872029 0.0000000       ***

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Value

A list containing the results of the regularized Bayesian estimation, which includes the model formula,dependent and context variables, and other relevant details from the analysis. The returned object is of class mlob_result.

Methods

The returned object supports the following S3 methods:

- print(x) - Display coefficients, standard errors, confidence intervals, Z-values, and p-values
- summary(x) - Comprehensive summary with significance stars and comparison to unoptimized ML
- coef(x) - Extract coefficients as a data frame
- se(x) - Extract standard errors
- vcov(x) - Extract variance-covariance matrix (diagonal only)
- confint(x, parm, level) - Extract confidence intervals for specified parameters
- as.data.frame(x) - Convert results to a data frame format

- `dim(x)` - Return dimensions (number of parameters)
- `length(x)` - Return number of parameters
- `names(x)` - Return parameter names
- `update(x, formula, data, conf.level, balancing.limit, punish.coeff, jackknife)` - Update model with new parameters

Author(s)

Valerii Dashuk <vadashuk@gmail.com>, Binayak Timilsina <binayak.timilsina001@gmail.com>, Martin Hecht, and Steffen Zitzmann

References

Dashuk, V., Hecht, M., Luedtke, O., Robitzsch, A., & Zitzmann, S. (2025a). doi:10.1017/psy.2025.10045

Dashuk, V., Hecht, M., Luedtke, O., Robitzsch, A., & Zitzmann, S. (2025b). doi:10.1007/s41237-025002647

Luedtke, O., Marsh, H. W., Robitzsch, A., Trautwein, U., Asparouhov, T., & Muthen, B. (2008). doi:10.1037/a0012869

Examples

```
# Example 1: usage with the iris dataset

result_iris <- mlob(
Sepal.Length ~ Sepal.Width + Petal.Length,
data = iris, group = 'Species',
conf.level = 0.99,
jackknife = FALSE)

# View summary statistics (similar to summary of a linear model);

summary(result_iris)

# Example 2: usage with highly unbalanced mtcars dataset (adjusted balancing.limit)

result_mtcars <- mlob(
mpg ~ hp + wt + am + hp:wt + hp:am,
data = mtcars, group = 'cyl',
balancing.limit = 0.35)

# View summary statistics

summary(result_mtcars)

# Example 3: Using all available S3 methods on slightly unbalanced ChickWeight dataset


result <- mlob(weight ~ Time, data = ChickWeight, group = 'Diet', jackknife = FALSE)

# Display methods
```

```
print(result)                    # Display results
summary(result)                 # Comprehensive summary
coef(result)                    # Extract coefficients
se(result)                      # Extract standard errors
vcov(result)                    # Extract variance-covariance matrix
confint(result)                 # Extract confidence intervals
confint(result, "beta_b")       # Extract CI for specific parameter
confint(result, level = 0.99)   # Extract CI with different confidence level
as.data.frame(result)            # Convert to data frame
dim(result)                     # Get dimensions
length(result)                  # Get number of parameters
names(result)                   # Get parameter names

# Update model with new parameters
update(result, conf.level = 0.99)

# List all available methods
methods(class = "mlob_result")
```

`names.mlob_result` *Get names of mlob_result objects*

Description

Get names of mlob_result objects

Usage

```
## S3 method for class 'mlob_result'
names(x)
```

Arguments

x	An object of class mlob_result

Value

The names of the parameters/coefficients

se *Generic function to extract standard errors*

Description

Generic function to extract standard errors

Usage

```
se(object, ...)
```

Arguments

object	An object
...	Additional arguments

se.mlob_result *Extract standard errors from mlob_result objects*

Description

Extract standard errors from mlob_result objects

Usage

```
## S3 method for class 'mlob_result'
se(object, ...)
```

Arguments

object	An object of class mlob_result
...	Additional arguments passed to se

Value

A named vector of standard errors

summary.mlob_result *Summary method for mlob_result objects*

Description

Summary method for mlob_result objects

Usage

```
## S3 method for class 'mlob_result'
summary(object, ...)
```

Arguments

object	An object of class mlob_result
...	Additional arguments passed to summary

update.mlob_result *Update mlob_result objects*

Description

Update mlob_result objects

Usage

```
## S3 method for class 'mlob_result'
update(
  object,
  formula = NULL,
  data = NULL,
  group = NULL,
  balancing.limit = NULL,
  conf.level = NULL,
  jackknife = NULL,
  punish.coeff = NULL,
  ...
)
```

Arguments

`object`	An object of class mlob_result
`formula`	Updated formula
`data`	Updated data
`group`	Updated group variable
`balancing.limit`	Updated balancing limit (default: 0.2)
`conf.level`	Updated confidence level (default: 0.95)
`jackknife`	Updated jackknife setting (default: FALSE)
`punish.coeff`	Updated punishment coefficient (default: 2)
`...`	Additional arguments passed to mlob

Value

Updated mlob_result object

`vcov.mlob_result` *Extract variances from mlob_result objects*

Description

Extract variances from mlob_result objects

Usage

```
## S3 method for class 'mlob_result'
vcov(object, ...)
```

Arguments

`object`	An object of class mlob_result
`...`	Additional arguments passed to vcov

Value

A named vector of variances (squared standard errors)

Index

as.data.frame, *5*
as.data.frame.mlob_result, 2

coef.mlob_result, 3
confint.mlob_result, 3

dim.mlob_result, 4

formula, *5*

length.mlob_result, 4

mlob, 5

names.mlob_result, 8

se, 9
se.mlob_result, 9
summary.mlob_result, 10

update.mlob_result, 10

vcov.mlob_result, 11