

Package ‘MultiLevelOptimalBayes’

July 21, 2025

Type Package

Title Regularized Bayesian Estimator for Two-Level Latent Variable Models

Version 0.0.2.0

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. Dashuk et al. (2024) <doi:10.13140/RG.2.2.18148.39048> derived the optimal regularized Bayesian estimator; Dashuk et al. (2024) <doi:10.13140/RG.2.2.34350.01604> extended it to the multivariate case; and Luedtke et al. (2008) <doi:10.1037/a0012869> formalized the two-level latent variable framework.

Imports pracma

License GPL-3

Depends R (>= 4.1.0)

Encoding UTF-8

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2025-07-11 22:20:07 UTC

Contents

mlob	2
Index	5

mlob	<i>Multi-Level Optimal Bayes Function (MLOB)</i>
------	--

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 = 0.05,  
  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., $Y \sim X + C \dots$), 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

<code>conf.level</code>	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.
<code>jackknife</code>	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.
<code>punish.coeff</code>	a multiplier that punishes the balancing procedure when deleting the whole group. If <code>punish.coeff</code> is equal to 1, no additional punishment is applied for deleting the group. Higher values intensify the penalty. Defaults to 2.
<code>...</code>	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)	
<code>beta_b</code>	0.4279681	0.7544766	-1.5154349	2.371371 0.5672384 0.57055223
<code>gamma_Petal.Length</code>	0.4679522	0.2582579	-0.1972762	1.133181 1.8119567 0.06999289

For comparison, summary of coefficients from unoptimized analysis (ML):

	Estimate	Std. Error	Lower CI (99)	
<code>beta_b</code>	0.6027440	5.424780e+15	-1.397331e+16	1.397331e+16 1.111094e-16 1.00000000
<code>gamma_Petal.Length</code>	0.4679522	2.582579e-01	-1.972762e-01	1.133181e+00 1.811957e+00 0.06999289

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.

Author(s)

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

References

Dashuk, V., Hecht, M., Luedtke, O., Robitzsch, A., & Zitzmann, S. (2024). [doi:10.13140/RG.2.2.18148.39048](https://doi.org/10.13140/RG.2.2.18148.39048)

Dashuk, V., Hecht, M., Luedtke, O., Robitzsch, A., & Zitzmann, S. (2024). [doi:10.13140/RG.2.2.34350.01604](https://doi.org/10.13140/RG.2.2.34350.01604)

Luedtke, O., Marsh, H. W., Robitzsch, A., Trautwein, U., Asparouhov, T., & Muthen, B. (2008). [doi:10.1037/a0012869](https://doi.org/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 = 0.01,
  jackknife = FALSE)

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

summary(result_iris)

# Example 2: usage with slightly unbalanced ChickWeight dataset

## Not run:
result_ChickWeight <- mlob(
  weight ~ Diet,
  data = ChickWeight,
  group = 'Time',
  punish.coeff = 1.5,
  jackknife = FALSE)

# View the results

print(result_ChickWeight)

# View summary statistics

summary(result_ChickWeight)

## End(Not run)

# Example 3: 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)
```

Index

`as.data.frame`, [2](#)

`formula`, [2](#)

`mlob`, [2](#)