Package: ProfileGLMM (via r-universe)

June 4, 2026

Type Package

Title Bayesian Profile Regression using Generalised Linear Mixed Models

Version 1.1.0

Description Implements a Bayesian profile regression using a generalized linear mixed model as output model. The package allows for binary (probit mixed model) and continuous (linear mixed model) outcomes and both continuous and categorical clustering variables. The package utilizes 'RcppArmadillo' and 'RcppDist' for high-performance statistical computing in C++. For more details see Amestoy & al. (2025) <doi:10.48550/arXiv.2510.08304>.

License GPL-2

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.3.2

LinkingTo Rcpp, RcppArmadillo, RcppDist

Imports Rcpp, LaplacesDemon, MCMCpack, Matrix, Spectrum, mvtnorm

Depends R (>= 3.5)

URL https://github.com/MatteoAmestoy/ProfileGLMM-package

BugReports https://github.com/MatteoAmestoy/ProfileGLMM-package/issues

Suggests knitr, rmarkdown

VignetteBuilder knitr

Config/pak/sysreqs libgmp3-dev make

Repository https://matteoamestoy.r-universe.dev

Date/Publication 2026-02-03 10:58:21 UTC

RemoteUrl https://github.com/matteoamestoy/profileglmm-package

RemoteRef HEAD

RemoteSha 2fc3c9e48abf07147a1d5461dbac626be44753db

Contents

`encodeCat` *One-Hot Encodes Factor Variables (FIRST Level as Reference)*

Description

This function takes a dataframe, identifies all columns of class `factor`, and converts them into **dummy variables** using one-hot encoding via `stats::model.matrix`. For each factor, the function explicitly removes the first dummy variable generated, effectively making the **first level** of the factor the **reference level** (omitted category). Non-factor columns are retained as is.

Usage

```
encodeCat(dataframe)
```

Arguments

`dataframe`	A `data.frame` containing the data to be processed, which may include factor variables.

Value

A `data.frame` where:

- All original non-factor columns are present.
- All original factor columns are replaced by a set of binary (0/1) dummy variables. The first level of the factor is excluded from the generated dummies, making the last level the reference.

Examples

```
data("exposure_data")
exp_data = exposure_data$df
covList = {}
covList$FE = c('X')
XFE = encodeCat(exp_data[,covList$FE, drop = FALSE])
```

`examp` *List of the different outputs of the main functions for the examples*

Description

A list of the different outputs of the main functions for the examples

Usage

```
examp
```

Format

A list with 4 components:

dataProfile Output of the profileGLMM_preprocess() function example

MCMC_Obj Output of the profileGLMM_Gibbs() function example

post_Obj Output of the profileGLMM_postprocess() function example

pred_Obj Output of the profileGLMM_predict() function example

Source

Generated synthetically by the package authors.

`exposure_data` *Simulated Data and Parameters for a exposure profile linear mixed model*

Description

A list containing a simulated exposure dataset (`df`) and the ground-truth parameters (`theta0`) used to generate it.

The dataset `df` contains $N = 4500$ observations across $n_{Ind} = 1500$ individuals, with $n_R = 3$ repeated measures per individual.

Usage

```
exposure_data
```

Format

A list with 2 components:

df A data frame with 4,500 rows and 6 variables (the simulated data).

theta0 A list of 11 elements containing the true parameters used for simulation.

Details

The underlying model for the response $\boldsymbol{Y}$ is:

$$\boldsymbol{Y} = \boldsymbol{X}_{Fe}\boldsymbol{\beta} + \boldsymbol{X}_{Int}\boldsymbol{\alpha}_{Lat} + \boldsymbol{X}_{Re}\boldsymbol{\alpha}_{RE} + \boldsymbol{\epsilon}$$

`df` Data Variables

X Continuous predictor ($\sim N(0, 1)$).

t Time-like variable (structured around 0, 1, 2).

indiv **Individual ID** (1 to 1500), the grouping factor.

Exp1, Exp2 Exposure continuous predictors.

Y The **Simulated Response Variable** calculated as: $\boldsymbol{Y} = y_{Fe} + y_{Int} + y_{Re} + \epsilon$, where $\epsilon\ N(0, 1)$.

`theta0` Parameters

The list `theta0` holds the true values used to generate `Y`, including:

- `Lat`: **Categorical Factor** (9 levels), defining the clusters for interaction effects.
- `beta`: True fixed effects for the global intercept and $\boldsymbol{X}$ (i.e., $(3, 2)$).
- `alphaLat`: Vector of 18 coefficients defining the cluster-specific intercepts and slopes for $\boldsymbol{X}$ within the 9 `Lat` categories.
- `alphaRE`: Vector of 1500 random slopes for the time variable $\boldsymbol{t}$, drawn from $N(0, 1)$.
- `sigma`: Residual standard deviation (1).

Source

Generated synthetically by the package authors.

`piecewise_data` *Simulated Data and Parameters for a Piecewise Example*

Description

A list containing a second simulated dataset (`df`) and its ground-truth parameters (`theta0`). This dataset is generated from a **piecewise linear model**, where the continuous predictor `x` is segmented into 6 bins, and different intercept and slope coefficients are applied to each segment.

The dataset `df` contains $N = 3000$ observations.

Usage

```
piecewise_data
```

Format

A list with 2 components:

df A data frame with 3,000 rows and 2 variables (the simulated data).

theta0 A list of 5 elements containing the true parameters used for simulation.

Details

The underlying model for the response $\boldsymbol{Y}$ is:

$$\boldsymbol{Y} = \boldsymbol{X}_{Fe}\boldsymbol{\beta} + \boldsymbol{X}_{Lat}\boldsymbol{\alpha}_{Lat} + \boldsymbol{\epsilon}$$

where $\boldsymbol{X}_{Fe}$ is the global intercept, and $\boldsymbol{X}_{Lat}\boldsymbol{\alpha}_{Lat}$ models the piecewise relationship of `x` across the 6 categories defined in `theta0$Lat`. The error term $\boldsymbol{\epsilon}$ $N(0, 1)$.

`df` Data Variables

x A continuous predictor, uniformly distributed between -3 and 3.

Y The **Simulated Response Variable** defined by the piecewise linear model.

`theta0` Parameters

The list `theta0` holds the true values used for simulation, including:

- `beta`: True global intercept (i.e., (0.5)).
- `Lat`: The categorical factor (1 to 6) derived from segmenting `x`.
- `alphaLat`: Vector of $2 * 6 = 12$ coefficients defining the specific intercept and slope for `x` within each of the 6 segments.

Source

Generated synthetically by the package authors.

`plot.pglmm_fit` *Plot method for pglmm_fit continuous covariates cluster characteristics*

Description

Plot method for pglmm_fit continuous covariates cluster characteristics

Usage

```
## S3 method for class 'pglmm_fit'
plot(x, ...)
```

Arguments

x	An object of class pglmm_fit
...	Additional arguments • title : main title of the plot • color : palette to be used

---

predict.pglmm_fit *Prediction of cluster memberships and outcomes*

Description

(This documentation is now for internal use only)

Usage

```
## S3 method for class 'pglmm_fit'
predict(object, newData, ...)
```

Arguments

object	An object of class pglmm_fit .
newData	: A list with fields • XFE A numeric matrix of fixed effects covariates for the prediction data. • XLat A numeric matrix of latent effect covariates. • UCont A numeric matrix or vector of continuous profile variables. Defaults to NULL. • UCat A numeric matrix or vector of categorical profile variables. Defaults to NULL.
...	Additional arguments

Examples

```
# Load post_Obj, the result of profileGLMM_postProcess()
data("examp")
post_Obj = examp$post_Obj

# run prediction for training data
pred_Obj = predict(post_Obj,examp$dataProfile$d)
```

print.pglmm_data *Print method for pglmm_data*

Description

Print method for pglmm_data

Usage

```
## S3 method for class 'pglmm_data'
print(x, ...)
```

Arguments

x	An object of class pglmm_data
...	Additional arguments

print.pglmm_fit *Print method for pglmm_fit*

Description

Print method for pglmm_fit

Print method for pglmm_fit

Usage

```
## S3 method for class 'pglmm_fit'
print(x, ...)

## S3 method for class 'pglmm_fit'
print(x, ...)
```

Arguments

x	An object of class pglmm_fit
...	Additional arguments

print.pglmm_mcmc *Print method for pglmm_mcmc*

Description

Print method for pglmm_mcmc

Usage

```
## S3 method for class 'pglmm_mcmc'
print(x, ...)
```

Arguments

x	An object of class pglmm_mcmc
...	Additional arguments

prior_init *Initialize the prior hyperparameters for the Profile GLMM*

Description

This function establishes the prior distributions for all parameters in the Profile GLMM. It sets up vague, non-informative priors (often using small precision/large variance or conjugate forms like Wishart/Dirichlet) for the fixed effects ($beta_{FE}$), residual variance (σ^2), random effects covariance (Σ_{RE}), latent effects covariance (Σ_{Lat}), cluster parameters (means and covariances), and the Dirichlet Process parameters (α).

Usage

```
prior_init(params)
```

Arguments

params	A list containing dimensional parameters of the model (often the output of process_Data_outcome). Important fields used for prior setup include: qFE: Number of fixed effects coefficients. qRE: Dimension of the random effects vector. qLat: Dimension of the latent effects vector. qUCont: Number of continuous profile variables. qUCat: Number of categorical profile variables.

Value

A list (prior) containing the hyperparameter values structured by the parameter block they govern:

FE: Priors for fixed effects and residual variance (e.g., lambda, a, b for conjugate Normal-Gamma).

RE: Inverse-Wishart priors for random effects covariance (Σ_{RE}) (e.g., Phi, eta).

assign: Priors for the cluster assignment parameters, nested under Cont (Normal-Inverse-Wishart for continuous) and Cat (Dirichlet for categorical).

Lat: Inverse-Wishart prior for the latent effects covariance (Σ_{Lat}) (e.g., Phi, eta).

DP: Parameters for the Dirichlet Process prior (e.g., scale, shape).

Examples

```
# Load dataProfile, the result of profileGLMM_preProcess()
data("examp")
dataProfile = examp$dataProfile
prior_config <- prior_init(dataProfile$params)
```

---

profileGLMM_Gibbs *R Wrapper for Profile GLMM Gibbs Sampler (C++ backend)*

Description

This is the main function for fitting the Profile Generalized Linear Mixed Model using a blocked Gibbs sampling algorithm. It acts as an R wrapper, passing an object of class pglmm_data directly to the RCPP implementation GSLoopCPP. The function simulates the posterior distribution of all model parameters, including fixed effects, random effects variance, profile cluster parameters, latent effects, and cluster assignments.

Usage

```
profileGLMM_Gibbs(model, nIt, nBurnIn)
```

Arguments

model	An object of class glmm_data (the output of profileGLMM_preprocess). This contains the design matrices, initial values, dimensions, and prior hyperparameters.
nIt	Integer, the total number of MCMC iterations counting the burn-in period. The sampler will return nIt - nBurnIn iterations in total.
nBurnIn	Integer, the number of initial MCMC iterations that are discarded (not saved) to allow the chain to converge.

Value

An object of class pglmm_mcmc. This is a list containing the saved Gibbs-sampled MCMC chains for all model parameters (e.g., beta, Z, gamma, pvec, muClus, PhiClus, etc.) and the variable names from the original data. This output is intended for post-processing with profileGLMM_postProcess.

Examples

```
# Load examp, which contains a pre-processed pglmm_data object
data("examp")
dataProfile = examp$dataProfile

# Run the Gibbs Sampler
MCMC_Obj = profileGLMM_Gibbs(
  model = dataProfile,
  nIt = 100,
  nBurnIn = 10
)
```

`profileGLMM_postProcess`

Post-process MCMC Output for Profile GLMM

Description

This function performs essential post-processing of the MCMC output generated by `profileGLMM_Gibbs`. It calculates posterior means and credible intervals for fixed effects and, optionally, computes a representative cluster partition using Least Squares (LS) or Ng's spectral clustering (NG). It also estimates cluster characteristics such as centroids, probability vectors, and outcome effects for the chosen partition.

Usage

```
profileGLMM_postProcess(
  MCMC_Obj,
  modeClus = "NG",
  comp_cooc = TRUE,
  alpha = 0.05
)
```

Arguments

`MCMC_Obj`	An object of class `pglmm_mcmc` (the output of `profileGLMM_Gibbs`).
`modeClus`	A character string specifying the clustering method. Options are `'NG'` (Ng's spectral clustering, default) or `'LS'` (Least Squares clustering).
`comp_cooc`	A logical value. If `TRUE` (default), the co-occurrence matrix is computed and clustering is performed. If `FALSE`, only the population parameters are processed.
`alpha`	A numeric value between 0 and 1, specifying the significance level for credible intervals. Defaults to `0.05` (95% CIs).

Value

An object of class pglmm_fit. This is a list containing:

- coocMat: The co-occurrence matrix of MCMC cluster assignments.
- clust: A list of representative clustering results (if comp_cooc = TRUE), including the optimal partition (Zstar), number of clusters (Kstar), and cluster-specific parameters (cen, pvec, gamma).
- pop: A list containing the posterior means and credible intervals for fixed effects.

Examples

```
# Load MCMC_Obj, the result of profileGLMM_Gibbs()
data("examp")
MCMC_Obj = examp$MCMC_Obj

# Post-process the results
post_Obj = profileGLMM_postProcess(MCMC_Obj, modeClus='LS')

# Removing the cooc matrix to save space
post_Obj$coocMat = NULL
```

---

profileGLMM_preprocess

Preprocess the data from a list describing the profile LMM model

---

Description

Preprocess the data from a list describing the profile LMM model

Usage

```
profileGLMM_preprocess(
  regType,
  covList,
  dataframe,
  nC,
  intercept = list(FE = TRUE, RE = TRUE, Lat = TRUE)
)
```

Arguments

regType	A string, current possibilities: linear or probit
covList	A list with fields: • FE fixed effect covariates names/index in dataframe • RE random effect covariates names/index in dataframe

- Lat latent effect covariates names/index in dataframe
- Assign assignement variables list with fields:
  - Cont Continuous variables names/index in dataframe
  - Cat Categorical variables names/index in dataframe
- REunit statistical unit of the RE colomn name/index
- Y outcome (Continuous)

`dataframe` A dataframe containing outcome anf covariates

`nC` int: maximal number of cluster for the DP truncation

`intercept` (optionnal): A list with fields

- RE bool indicating if FE have an intercept
- FE bool indicating if RE have an intercept
- Lat bool indicating if Latent have an intercept

Value

An object of class `pglmm_data`. This is a list with

- d dictionary with [XFE,XRE,XLat,UCont,UCat,ZRE] design matrices
- [[params]] list of the parameters of the data
  - n int nb of obs
  - qFE int, number of covariates of FE
  - nRE int, number of stat units of RE
  - qRE int, number of covariates of RE
  - qLat int, number of covariates interacting with the latent clusters
  - qUCont int, number of continuous clustering covariates
  - qUCat int, number of categorical clustering covariates
  - nC int, maximal number of clusters
- prior a list with all the specification of the default prior used
- theta a list with a default set of parameters to start the chain, drawn from the prior
- regType an int. Currently 0 for linear, 1 for probit

Examples

```
data("exposure_data")
exp_data = exposure_data$df
theta0 = exposure_data$theta0
covList = {}
covList$FE = c('X')
covList$RE = c('t')
covList$REunit = c('indiv')

covList$Lat = c('X')

covList$Assign$Cont = c('Exp1','Exp2')
covList$Assign$Cat = NULL
```

```
covList$Y = c('Y')
dataProfile = profileGLMM_preprocess(regType = 'linear',
                                     covList = covList,
                                     dataframe = exp_data,
                                     nC = 30,
                                    intercept = list(FE = TRUE, RE = FALSE, Lat = TRUE))
```

`summary.pglmm_fit` *Print method for pglmm_fit*

Description

Print method for pglmm_fit

Usage

```
## S3 method for class 'pglmm_fit'
summary(x, ...)
```

Arguments

`x`	An object of class `pglmm_fit`
`...`	Additional arguments

`theta_init` *Initialize the variables for the Gibbs sampler chain*

Description

This function generates initial values (`theta`) for all parameters in the Profile GLMM Gibbs sampler by drawing from the specified prior distributions. These initial values are crucial for starting the MCMC chain in `profileGLMM_Gibbs`. The initialization includes parameters for fixed effects, random effects variance, latent effects, and the profile cluster parameters (centroids, covariances, and categorical probability vectors).

Usage

```
theta_init(prior, params)
```

Arguments

`prior`	A list containing the prior configuration to draw initialization from. This list should match the structure produced by the `prior_init` function, including hyperparameters for FE, RE, Latent, and cluster assignment priors.
`params`	A list containing the problem's dimensional parameters and indices (e.g., number of observations, number of covariates). This list should match the structure of the output from `process_Data_outcome`.

Value

A list (`theta`) containing the sampled initialization values for the Gibbs sampler. Key elements include:

`sig2`: Initial residual variance.

`betaFE`: Initial fixed effects coefficients.

`SigRE`: Initial random effects covariance matrix.

`SigLat`: Initial latent effects covariance matrix.

`gammaLat`: Initial latent effects coefficients, organized by cluster.

`ClusCont`: List containing initial continuous cluster parameters (`mu` and `Sigma`).

`ClusCat`: List containing initial categorical cluster parameters (`pvecClus`).

Examples

```
# Load dataProfile, the result of profileGLMM_preProcess()
data("examp")
dataProfile = examp$dataProfile
theta = theta_init(dataProfile$prior,dataProfile$params)
```

Index