Package ‘multifamm’

September 28, 2021

Type Package

Title Multivariate Functional Additive Mixed Models

Version 0.1.1

Description An implementation for multivariate functional additive mixed models (multiFAMM), see Volkmann et al. (2021, <arXiv:2103.06606>). It builds on developed methods for univariate sparse
functional regression models and multivariate functional principal component analysis. This package contains the function to run a multiFAMM and some convenience functions useful when working with large models. An additional package on GitHub contains more convenience functions to reproduce the analyses of the corresponding paper (<https://github.com/alexvolkmann/multifammPaper>).

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends R (>= 3.5.0)

Imports data.table, funData, MFPCA (>= 1.3-2), mgcv, sparseFLMM (> 0.3.0), stats, zoo

NeedsCompilation no

Author Alexander Volkmann [aut, cre]

Maintainer Alexander Volkmann <alexandervolkmann8@gmail.com>

Repository CRAN

Date/Publication 2021-09-28 09:00:02 UTC

R topics documented:

`extract_components` *Extract Model Components to be Compared*

Description

This is an internal function that helps to compare different models. The models resulting from a multiFAMM() call are typically very big. This function extracts the main information from a model so that a smaller R object can be saved.

Usage

```
extract_components(model, dimnames)
```

Arguments

`model`	multiFAMM model object from which to extract the information.
`dimnames`	Names of the dimensions of the model.

Details

So far the grid is fixed to be on [0,1].

Value

A list with the following elements

- `error_var`: A list containing the following elements
  - `model_weights`: Model weights used in the final multiFAMM.
  - `modelsig2`: Estimate of sigma squared in the final model.
  - `uni_vars`: Univariate estimates of sigma squared.
- `eigenvals`: List containing the estimated eigenvalues.
- `fitted_curves`: multiFunData object containing the fitted curves.
- `eigenfcts`: multiFunData object containing the estimated eigenfunctions.
- `cov_preds`: multiFunData object containing the estimated covariate effects.
- `ran_preds`: List containing multiFunData objects of the predicted random effects.
- `scores`: List containing matrices of the estimated scores.
- `meanfun`: multiFunData object containing the estimated mean function.
- `var_info`: List containing all eigenvalues and univariate norms before the MFPC pruning step
  - `eigenvals`: Vector of all multivariate eigenvalues.
  - `uni_norms`: List of univariate norms of all eigenfunctions.

extract_components_uni
Extract Model Components to be Compared from Univariate Model

Description

This is an internal function that helps to compare different models. The models resulting from a multiFAMM() call are typically very big. This function extracts the main information from a univariate model so that a smaller R object can be saved.

Usage

```
extract_components_uni(model)
```

Arguments

`model`	Univariate multiFAMM model object from which to extract the information.

Details

So far the grid is fixed to be on [0,1].

Value

A list with the following elements

- `error_var`: A list containing the following elements
  - `model_weights`: Model weights used in the final multiFAMM.
  - `modelsig2`: Estimate of sigma squared in the final model.
  - `uni_vars`: Univariate estimates of sigma squared.
- `eigenvals`: List containing the estimated eigenvalues.
- `fitted_curves`: multiFunData object containing the fitted curves.
- `eigenfcts`: multiFunData object containing the estimated eigenfunctions.
- `cov_preds`: multiFunData object containing the estimated covariate effects.
- `ran_preds`: List containing multiFunData objects of the predicted random effects.
- `scores`: List containing matrices of the estimated scores.

multiFAMM *Multivariate Functional Additive Mixed Model Regression*

Description

This is the main function of the package and fits the multivariate functional additive regression model with potentially nested or crossed functional random intercepts.

Usage

```
multiFAMM(data, fRI_B = FALSE, fRI_C = FALSE, nested = FALSE,
  bs = "ps", bf_mean = 8, bf_covariates = 8, m_mean = c(2, 3),
  covariate = FALSE, num_covariates = NULL, covariate_form = NULL,
  interaction = FALSE, which_interaction = matrix(NA), bf_covs, m_covs,
  var_level = 1, use_famm = FALSE, save_model_famm = FALSE,
  one_dim = NULL, mfpc_weight = FALSE, mfpc_cutoff = 0.95,
  number_mfpc = NULL, mfpc_cut_method = c("total_var", "unidim"),
  final_method = c("w_bam", "bam", "gaulss"), weight_refit = FALSE,
  verbose = TRUE, ...)
```

Arguments

data	Data.table that contains the information with some fixed variable names, see Details.
fRI_B	Boolean for including functional random intercept for individual (B in Cederbaum). Defaults to FALSE.
fRI_C	Boolean for including functional random intercept for word (C in Cederbaum). Defaults to FALSE.
nested	TRUE to specify a model with nested functional random intercepts for the first and second grouping variable and a smooth error curve. Defaults to FALSE.
bs	Spline basis function, only tested for "ps" (as in sparseFLMM).
bf_mean	Basis dimension for functional intercept (as in sparseFLMM).
bf_covariates	Basis dimension for all covariates (as in sparseFLMM).
m_mean	Order of penalty for basis function (as in sparseFLMM).
covariate	Covariate effects (as in sparseFLMM).
num_covariates	Number of covariates included in the model (as in sparseFLMM).
covariate_form	Vector of strings for type of covariate (as in sparseFLMM).
interaction	TRUE if there are interactions between covariates (as in sparseFLMM). Defaults to FALSE.
which_interaction	Symmetric matrix specifying the interaction terms (as in sparseFLMM).
bf_covs	Vector of marginal basis dimensions for fRI covariance estimation (as in sparse-FLMM).

- `m_covs` List of marginal orders for the penalty in fRI covariance estimation (as in sparseFLMM).
- `var_level` Pre-specified level of explained variance on each dimension (as in sparseFLMM). Defaults to including all non-negative Eigenvalues.
- `use_famm` Re-estimate the mean in FAMM context (as in sparseFLMM) - overwritten by one_dim.
- `save_model_famm` Give out the FAMM model object (as in sparseFLMM) - overwritten by one_dim.
- `one_dim` Specify the name of the dimension if sparseFLMM is to be computed only on one dimension.
- `mfpc_weight` TRUE if the estimated univariate error variance is to be used as weights in the scalar product of the MFPCA.
- `mfpc_cutoff` Pre-specified level of explained variance of results of MFPCA. Defaults to 0.95.
- `number_mfpc` List containing the number of mfPCs needed for each variance component e.g. list("E" = x, "B" = y).
- `mfpc_cut_method` Method to determine the level of explained variance
  - total_var: (weighted) sum of variation over the dimensions.
  - unidim: separate on each dimension.
- `final_method` Function used for estimation of final model to allow for potential heteroscedasticity ("w_bam", "bam", "gaulss").
- `weight_refit` Get the weights for the weighted bam by first refitting the model under an independence assumption but with mfpc basis functions. Defaults to FALSE.
- `verbose` Print progress of the multifamm. Defaults to TRUE.
- `...` Additional arguments to be passed to (mainly) the underlying sparseFLMM function.

Details

Expand the method proposed by Fabian Scheipl to incorporate the variance decomposition developed by Cederbaum et al. (2016). To account for the correlation between the dimensions, the MFPCA approach by Happ and Greven (2016) is applied.

The data set has to be of the following format:

- y_vec (numeric): vector of response values
- t (numeric): observation point locations
- n_long (integer): curve identification
- subject_long (integer): subject identification (NEEDS TO BE SPECIFIED)
- word_long (integer): word identification
- combi_long (integer): repetition
- dim (factor): level of the dimension
- covariate.1 (numeric): potential covariate(s) named with trailing 1,2,3,...

It is possible to introduce weights for the final estimation of the multiFAMM. Currently, it is only implemented to use the inverse of the univariate measurement error estimates as weights. Note that negative values of variance estimates are set to zero in fast symmetric additive covariance smoothing. In order to still include weights, zero-values are substituted by values of the smallest positive variance estimate.

Value

A list with five elements

- the final multivariate FAMM
- the sparseFLMM output for each of the dimensions
- information on the untruncated MPFCA results
- the truncated MFPC output
- the data used to fit the model.

Examples

```
# subset of the phonetic data (very small subset, no meaningful results can
# be expected and no random effects other than the random smooth should be
# included in the model)

data(phonetic_subset)

m <- multiFAMM(data = phonetic_subset, covariate = TRUE, num_covariates = 2,
               covariate_form = c("by", "by"), interaction = TRUE,
               which_interaction = matrix(c(FALSE, TRUE, TRUE, FALSE),
               nrow = 2, ncol = 2), bf_covs = c(5), m_covs = list(c(2, 3)),
               mfpc_cut_method = "total_var", final_method = "w_bam")
```

`phonetic` *Phonetic data*

Description

The data are part of a large study on consonant assimilation, which is the phenomenon that the articulation of two consonants becomes phonetically more alike when they appear subsequently in fluent speech. The data set contains the audio signals of nine different speakers which repeated the same sixteen German target words each five times. In addition to these acoustic signals, the data set also contains the electropalatographic data. The target words are bisyllabic noun-noun compound words which contained the two abutting consonants of interest, s and sh, in either order. Consonant assimilation is accompanied by a complex interplay of language-specific, perceptual and articulatory factors. The aim in the study was to investigate the assimilation of the two consonants as a function of their order (either first s, then sh or vice-versa), syllable stress (stressed or unstressed) and vowel context, i.e. which vowels are immediately adjacent to the target consonants of interest. The vowels are either of the form ia or ai. For more details, see references below.

Usage

```
phonetic
```

Format

A data.frame with 50644 observations and 12 variables:

`dim` Factor for identifying the acoustic (aco) and electropalatographic (epg) dimensions.

`subject_long` Unique identification number for each speaker.

`word_long` Unique identification number for each target word.

`combi_long` Number of the repetition of the combination of the corresponding speaker and target word.

`y_vec` The response values for each observation point.

`n_long` Unique identification number for each curve.

`t` The observations point locations.

`covariate.1` Order of the consonants, reference category first /s/ then /sh/.

`covariate.2` Stress of the final syllable of the first compound, reference category 'stressed'.

`covariate.3` Stress of the initial syllable of the second compound, reference category 'stressed'.

`covariate.4` Vowel context, reference category ia.

`word_names_long` Names of the target words

Source

Pouplier, Marianne and Hoole, Philip (2016): Articulatory and Acoustic Characteristics of German Fricative Clusters, Phonetica, 73(1), 52–78.

Cederbaum, Pouplier, Hoole, Greven (2016): Functional Linear Mixed Models for Irregularly or Sparsely Sampled Data. Statistical Modelling, 16(1), 67-88.

Jona Cederbaum (2019). sparseFLMM: Functional Linear Mixed Models for Irregularly or Sparsely Sampled Data. R package version 0.3.0. https://CRAN.R-project.org/package=sparseFLMM

`phonetic_subset` *Phonetic data (subset)*

Description

A small subset of the phonetics data set `phonetic` with observations from two speakers and two items only. This will not produce meaningful results but can be used as a toy data set when testing the code. The variables are as in the full data set, see `phonetic`

Usage

```
phonetic_subset
```

Format

A data.frame with 1336 observations and 12 variables.

Source

Pouplier, Marianne and Hoole, Philip (2016): Articulatory and Acoustic Characteristics of German Fricative Clusters, Phonetica, 73(1), 52–78.

Cederbaum, Pouplier, Hoole, Greven (2016): Functional Linear Mixed Models for Irregularly or Sparsely Sampled Data. Statistical Modelling, 16(1), 67-88.

Jona Cederbaum (2019). sparseFLMM: Functional Linear Mixed Models for Irregularly or Sparsely Sampled Data. R package version 0.3.0. https://CRAN.R-project.org/package=sparseFLMM

`snooker` *Snooker data*

Description

The data are part of a study on the impact of a muscular training program on snooker technique. 25 recreational snooker players were split into treatment (receiving instructions for a training program) and control group (no training program). The data set contains the movement trajectories of the snooker players in two sessions (before and after the training period), where each snooker player repeated a snooker shot of maximal force six times. The interest lies in the movement of hand, elbow, and shoulder on a two-dimensional grid (called X and Y). The trajectories are normalized on a [0,1] time grid and the beginning of the hand trajectories are centered to the origin.

Usage

```
snooker
```

Format

A data.frame with 56910 observations and 11 variables:

`y_vec` The response values for each observation point.

`t` The observations point locations.

`n_long` Unique identification number for each curve.

`subject_long` Unique identification number for each snooker player.

`word_long` Integer specifying the session. 1: Before the training, 2: After the training.

`dim` Factor for identifying the univariate dimensions.

`combi_long` Number of the repetition of the snooker shot.

`covariate.1` Skill level of the snooker player. 0: Unskilled, 1: Skilled.

`covariate.2` Group of the snooker player. 0: Control group, 1: Treatment group.

`covariate.3` Session indicator. 0: Before the treatment, 1: After the treatment.

`covariate.4` Interaction of group and session, i.e. the treatment effect indicator.

Source

Enghofer, T. (2014). Überblick über die Sportart Snooker, Entwicklung eines Muskeltrainings und Untersuchung dessen Einflusses auf die Stoßtechnik. Unpublished Zulassungsarbeit, Technische Universität München.

Index