Package: grove (via r-universe)

October 27, 2026

Type Package

Title Wavelet Functional ANOVA Through Markov Groves

Version 1.1.1

Date 2023-12-09

Author Jacopo Soriano and Li Ma

Maintainer Li Ma <li.ma@duke.edu>

Description Functional denoising and functional ANOVA through wavelet-domain Markov groves. Fore more details see: Ma L. and Soriano J. (2018) Efficient functional ANOVA through wavelet-domain Markov groves. <doi:10.48550/arXiv.1602.03990v2>.

License GPL (>= 3)

Imports methods, Rcpp (>= 0.11.2), wavethresh

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

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

Date/Publication 2023-12-11 02:40:00 UTC

RemoteUrl https://github.com/cran/grove

RemoteRef HEAD

RemoteSha e6081aaf15ca47c93e667dcb55ed528dc2d96371

Contents

Denoise 2
DWT 3
FAnova 3
GenerateSyntheticAnova 4
grove 5
InvDWT 5
PlotFun 6
PlotStates 7

Index **8**

`Denoise` *Bayesian wavelet denoising*

Description

This function carries out Bayesian wavelet denoising using the Normal Inverse Gamma Markov Tree method of Ma and Soriano (2016).

Usage

```
Denoise(W, alpha = 0.5, nu = 5, n.samples = 500,
  transition.mode = "Markov", method = "Nelder-Mead")
```

Arguments

`W`	An object of class `DWT`.
`alpha`	Hyperparameter controlling the global smoothness.
`nu`	Hyperparameter controlling variance heterogeneity. If `Inf`, then the variance is identical for all nodes.
`n.samples`	Number of posterior draws.
`transition.mode`	Type of transition. The two options are `Markov` or `Independent`.
`method`	Method used for find maxmimum of marginal likelihood.

Value

An object of class `grove`.

References

Ma L. and Soriano J. (2016) Efficient functional ANOVA through wavelet-domain Markov groves. arXiv:1602.03990v2 [stat.ME] (https://arxiv.org/abs/1602.03990v2).

Examples

```
data <- wavethresh::DJ.EX(n = 512, noisy = TRUE, rsnr = 5)$doppler
W <- DWT(data)
ans <- Denoise(W)
```

DWT *Discrete wavelet transform*

Description

This function performs the discrete wavelet transform (DWT) according to Mallat's pyramidal algorithm (Mallat, 1989).

Usage

```
DWT(data, filter.number = 10, family = "DaubLeAsymm")
```

Arguments

`data`	A matrix of data, where each row is an observation. The number of columns must be a power of two.
`filter.number`	The smoothness of the wavelet to use in the decomposition.
`family`	The family of wavelets. The two most common options are `DaubExPhase` and `DaubLeAsymm`.

Details

See function `wd` from package `wavethresh` for more details.

Value

A `DWT` object. This object is a list with the following components:

Examples

```
data <- GenerateSyntheticAnova(st.dev = 5, n.replicates = 10)
W <- DWT(data$noisy.Y)
```

FAnova *Bayesian functional ANOVA*

Description

This function carries out Bayesian functional ANOVA using the Normal Inverse Gamma Markov Grove method of Ma and Soriano (2016).

Usage

```
FAnova(W, X, formula, nu = 5, is.kappa.fixed = FALSE, gamma.kappa = 0.3,
  eta.kappa = 0.1, n.samples = 500, transition.mode = "Markov",
  method = "Nelder-Mead")
```

Arguments

W	An object of class DWT.
X	Design matrix.
formula	An object of class formula.
nu	Hyperparameter controlling the heterogeneity in the noise variance.
is.kappa.fixed	If TRUE, gamma.kappa and eta.kappa are fixed. If FALSE gamma_kappa and eta_kappa are determined using Empirical Bayes.
gamma.kappa	Hyperparameter for the MT transition matrix.
eta.kappa	Hyperparameter for the MT transition matrix.
n.samples	Number of posterior draws.
transition.mode	Type of transition. The two options are Markov or Independent.
method	Method used for find maxmimum of marginal likelihood.

Value

An object of class grove.

References

Ma L. and Soriano J. (2016) Efficient functional ANOVA through wavelet-domain Markov groves. arXiv:1602.03990v2 [stat.ME] (https://arxiv.org/abs/1602.03990v2).

Examples

```
## Not run:
data <- GenerateSyntheticAnova(st.dev = 5, n.replicates = 5)
W <- DWT(data$noisy.Y)
X <- data$X
ans <- FAnova(W, X, ~ 1 + factorA + factorB)
denoised.data <- InvDWT(ans, x = c(0, 0, 1, 0))
PlotFun(denoised.data)
## End(Not run)
```

GenerateSyntheticAnova

Generate synthetic functional ANOVA dataset

Description

This function generates a synthetic 3-factor functional ANOVA dataset.

Usage

```
GenerateSyntheticAnova(st.dev = 10, n.replicates = 5)
```

Arguments

st.dev	The standard deviation of the error.
n.replicates	The number of replicates for each factor combination.

Value

A list containing the data without noise, the data with noise, and the design matrix.

Examples

```
data <- GenerateSyntheticAnova(st.dev = 5, n.replicates = 10)
ix <- 1
plot(data$clean.Y[ix, ], type = "l", col = "red", ylab = "")
lines(data$noisy.Y[ix, ], col = "blue")
```

---

grove *grove: A package for functional denoising and functional ANOVA*

Description

The grove package implements a wavelet-domain Bayesian hierarchical model for functional analysis of variance.

InvDWT *Inverse discrete wavelet transform*

Description

This function performs the inverse discrete wavelet transform.

Usage

```
InvDWT(grove.obj, x = NULL, include.C = TRUE, sample.C = FALSE)
```

Arguments

grove.obj	An object of class grove.
x	A vector of the values of a predictor.
include.C	If TRUE, C is used for reconstructing the function.
sample.C	If TRUE, draws from C are used for recontructing the function.

Value

A matrix with each row representing a draw from the reconstructed signal.

Examples

```
data <- wavethresh::DJ.EX(n = 512, noisy = TRUE, rsnr = 5)$doppler
W <- DWT(data)
ans <- Denoise(W)
denoised.data <- InvDWT(ans)
plot(data, type = "l")
lines(denoised.data[1, ], col = "red")
```

`PlotFun` *Function to plot the denoised signal*

Description

This function plots the credible bounds of the denoised signal.

Usage

```
PlotFun(data, p = c(0.025, 0.5, 0.975), band.type = "pointwise",
  main = "", col = "blue", type = "l", ylab = "", xlab = "",
  ylim = NULL)
```

Arguments

`data`	Matrix of posterior samples.
`p`	Vector with the lower, center and upper quantile.
`band.type`	Type of credible intervals. The options are: `pointwise`, `gloabl` or `global`.
`main`	The main title of the plot.
`col`	The color of the point estimate.
`type`	The type of line of the point estimate.
`ylab`	The label of the y-axis.
`xlab`	The label of the x-axis.
`ylim`	The range of the y-axis.

Value

A plot.

Examples

```
data <- wavethresh::DJ.EX(n = 512, noisy = TRUE, rsnr = 5)$doppler
W <- DWT(data)
ans <- Denoise(W)
denoised.data <- InvDWT(ans)
PlotFun(denoised.data)
PlotFun(denoised.data, band.type = "both")
```

PlotStates *Function to plot the hidden states*

Description

This function plots on a tree the state of each latent variables.

Usage

```
PlotStates(grove.obj, block = "Intercept", legend = FALSE, main = NULL,
  prior = FALSE)
```

Arguments

grove.obj	Output from function FAnova.
block	Which block to plot.
legend	If TRUE, show legend.
main	Main title.
prior	If TRUE, plot prior state probabilities. If FALE, plot posterior state probabilities.

Value

A plot.

Examples

```
## Not run:
data <- GenerateSyntheticAnova(st.dev = 5, n.replicates = 5)
W <- DWT(data$noisy.Y)
X <- data$X
ans <- FAnova(W, X, ~ 1 + factorA + factorB)
PlotStates(ans)
PlotStates(ans, block = "factorA")
PlotStates(ans, block = "factorB")
## End(Not run)
```

Index

Denoise, 2
DWT, 3

FAnova, 3

GenerateSyntheticAnova, 4
grove, 5
grove-package *(grove)*, 5

InvDWT, 5

PlotFun, 6
PlotStates, 7