Package: fda.vi (via r-universe)

June 21, 2026

Title Functional Data Analysis using Variational Inference

Version 1.0.0

Maintainer Camila de Souza <camila.souza@uwo.ca>

Description Implements a variational Expectation-Maximization (VEM) algorithm for smoothing one or multiple functional observations via basis function selection. The algorithm estimates all model parameters simultaneously and automatically, while accounting for within-curve correlation. The approach provides a flexible and computationally efficient framework for smoothing correlated functional data.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.1)

Imports stats, graphics, fda, MASS, scales

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

LazyData true

VignetteBuilder knitr

URL https://github.com/desouzalab/fda.vi

BugReports https://github.com/desouzalab/fda.vi/issues

Config/pak/sysreqs make

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

Date/Publication 2026-04-14 21:07:02 UTC

RemoteUrl https://github.com/desouzalab/fda.vi

RemoteRef HEAD

RemoteSha 7bc5fb3cd79e72bb87d234841a3a706dd57560b1

Contents

`coef.vem_fit` *Extract Active Basis Coefficients from a VEM Fit*

Description

Returns a $K \times m$ matrix of estimated basis coefficients. Each column corresponds to one curve; each row to one basis function. Coefficients are set to zero when the posterior inclusion probability $p_{ki} \leq$ `threshold` (inactive bases). When `is.composite = TRUE`, the matrix has dimension $\max(K) \times m$, where $\max(K)$ is the highest K selected by GCV across all curves; coefficients for curves with smaller optimal K are zero-padded (structural padding).

Usage

```
## S3 method for class 'vem_fit'
coef(object, threshold = 0.5, ...)
```

Arguments

`object`	A `vem_fit` object from `vem_fit`.
`threshold`	Numeric in $(0, 1)$. Posterior inclusion probability below which a coefficient is set to zero. Default `0.5`.
`...`	Currently unused.

Value

A numeric matrix of dimension $\max(K) \times m$, with row names `B1`, `B2`, ... and column names `Curve_1`, `Curve_2`, ...

References

da Cruz, A. C., de Souza, C. P. E., & Sousa, P. H. T. O. (2024). Fast Bayesian basis selection for functional data representation with correlated errors. *arXiv:2405.20758*. https://arxiv.org/abs/2405.20758

See Also

vem_fit, predict.vem_fit, summary.vem_fit

Examples

```
data(toy_curves)
fit <- vem_fit(y = toy_curves$y, Xt = toy_curves$Xt, K = 8)

# K x m matrix of active coefficients
coefs <- coef(fit)
dim(coefs)   # 8 x 3

# Compare estimated vs true coefficients for curve 1
cbind(estimated = coefs[, 1], true = toy_curves$true_coef)

# Stricter threshold — only very confident inclusions
coef(fit, threshold = 0.9)
```

---

gcv_vem *GCV Score for a VEM Smooth Fit*

Description

Computes the generalized cross-validation (GCV) score for each curve from a vem_smooth model object. GCV approximates leave-one-out prediction error and is used by tune_vem_by_gcv to select the optimal number of basis functions K.

The smoother matrix S_i maps observed values to fitted values and is constructed from the variational posteriors. Its trace provides the effective degrees of freedom used in the GCV penalty.

Usage

```
gcv_vem(out, threshold = 0.5)
```

Arguments

out	A fitted object returned by vem_smooth.
threshold	Numeric in $(0, 1)$. Posterior inclusion probability threshold for treating a basis as active. Default 0.5.

Value

A named numeric vector of length m of per-curve GCV scores. Lower scores indicate better fit relative to model complexity.

References

da Cruz, A. C., de Souza, C. P. E., & Sousa, P. H. T. O. (2024). Fast Bayesian basis selection for functional data representation with correlated errors. *arXiv:2405.20758*. https://arxiv.org/abs/2405.20758

See Also

tune_vem_by_gcv, vem_smooth

`plot.vem_fit` *Plot a VEM Fit with Credible Band*

Description

Plots observed data, the posterior mean fitted curve, and an optional 95% credible band for a single curve from a `vem_fit` object. The credible band provides uncertainty quantification by sampling from the variational posteriors: $\beta_i \sim \text{MVN}(\boldsymbol{\mu}_{\boldsymbol{\beta}_i}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}_i})$ and $Z_{ki} \sim \text{Bernoulli}(p_{ki})$. Predictions are automatically back-transformed if the model was fitted with `center = TRUE` or `scale = TRUE`.

Usage

```
## S3 method for class 'vem_fit'
plot(
  x,
  curve_idx = 1,
  type = c("polygon", "lines"),
  show_CI = TRUE,
  n_samples = 200,
  alpha_shade = 0.25,
  ylim = NULL,
  xlab = "t",
  ylab = "Value",
  show_basis = FALSE,
  ...
)
```

Arguments

`x`	A `vem_fit` object from `vem_fit`.
`curve_idx`	Integer. Index of the curve to plot. Default 1.
`type`	Character. Credible band style: `"polygon"` (shaded region) or `"lines"` (dashed lines). Default `"polygon"`.
`show_CI`	Logical. If `TRUE`, compute and display the credible band. Default `TRUE`.
`n_samples`	Integer. Number of posterior draws used to construct the credible band. Default `200`.

alpha_shade	Numeric in $(0, 1)$. Opacity of the shaded credible band (type = "polygon" only). Default 0.25.
ylim	Optional numeric vector of length 2. If NULL, axis limits are set to cover the data and credible band.
xlab	Character. Label for the horizontal axis. Default "t" (the evaluation point). Supply a domain-specific label where appropriate (e.g., "Time", "Age", "Frequency").
ylab	Character. Label for the vertical axis. Default "Value".
show_basis	Logical. If TRUE, adds a subplot below the main plot showing each basis function coloured by inclusion status (blue = active, grey = inactive). Default FALSE.
...	Additional graphical parameters passed to plot().

Value

Invisibly returns NULL. Called for its side effect of producing a plot.

References

da Cruz, A. C., de Souza, C. P. E., & Sousa, P. H. T. O. (2024). Fast Bayesian basis selection for functional data representation with correlated errors. *arXiv:2405.20758*. https://arxiv.org/abs/2405.20758

See Also

vem_fit, predict.vem_fit

Examples

```
data(toy_curves)
fit <- vem_fit(y = toy_curves$y, Xt = toy_curves$Xt, K = 8)

# Default: shaded credible band for curve 1
plot(fit)

# Dashed credible band for curve 2
plot(fit, curve_idx = 2, type = "lines")

# With basis selection subplot
plot(fit, curve_idx = 1, show_basis = TRUE)

# Suppress credible band
plot(fit, show_CI = FALSE, main = "Mean fit only")
```

predict.vem_fit *Predict Method for VEM Fits*

Description

Returns posterior mean curve estimates from a `vem_fit` object. Active basis functions are selected by applying a 0.5 probability threshold on the posterior inclusion probabilities. If `newdata` is supplied, a new basis matrix is constructed at those time points; otherwise the original fitted time points are used. Predictions are automatically back-transformed if the model was fitted with `center = TRUE` or `scale = TRUE`.

Usage

```
## S3 method for class 'vem_fit'
predict(object, newdata = NULL, ...)
```

Arguments

`object`	A `vem_fit` object from `vem_fit`.
`newdata`	Optional numeric vector of new time points at which to evaluate the fitted curves. Must lie within the original domain `range(Xt)`. If `NULL`, predictions are returned at the original `Xt`.
`...`	Currently unused.

Value

A list of length m. Each element is a numeric vector of predicted values on the original (back-transformed) scale.

References

da Cruz, A. C., de Souza, C. P. E., & Sousa, P. H. T. O. (2024). Fast Bayesian basis selection for functional data representation with correlated errors. *arXiv:2405.20758*. https://arxiv.org/abs/2405.20758

See Also

`vem_fit`, `plot.vem_fit`, `coef.vem_fit`

Examples

```
data(toy_curves)
fit <- vem_fit(y = toy_curves$y, Xt = toy_curves$Xt, K = 8)

# Predictions at original time points
preds <- predict(fit)
length(preds)       # 3 — one vector per curve
```

```
# Predictions at a denser grid
Xt_new <- seq(0, 1, length.out = 200)
preds_dense <- predict(fit, newdata = Xt_new)

# Plot observed vs predicted for curve 1
plot(toy_curves$Xt, toy_curves$y[[1]],
     pch = 16, cex = 0.6, col = "grey50",
     xlab = "t", ylab = "y")
lines(Xt_new, preds_dense[[1]], col = "firebrick", lwd = 2)
```

`summary.vem_fit` *Summary Method for VEM Fits*

Description

Provides a displayed summary of the results from `vem_fit` and invisibly returns a list of summary statistics, including the basis type, number of curves, selected K, active basis counts per curve, estimated model parameters, and GCV tuning results if applicable.

Reported variational posterior parameters for σ^2 and τ^2 are the shape and scale of their respective Inverse-Gamma variational distributions: $q(\sigma^2) = \text{IG}(\delta_1^*, \delta_2^*)$ and $q(\tau^2) = \text{IG}(\lambda_1^*, \lambda_2^*)$. For composite fits (`selection_metric = "per_curve"`), parameters from the first curve are shown as representative values.

Usage

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

Arguments

`object`	A `vem_fit` object from `vem_fit`.
`...`	Currently unused.

Value

Invisibly returns a list with element `active_bases`: an integer vector of active basis counts per curve.

References

da Cruz, A. C., de Souza, C. P. E., & Sousa, P. H. T. O. (2024). Fast Bayesian basis selection for functional data representation with correlated errors. *arXiv:2405.20758*. https://arxiv.org/abs/2405.20758

See Also

`vem_fit`, `coef.vem_fit`

Examples

```
data(toy_curves)
fit <- vem_fit(y = toy_curves$y, Xt = toy_curves$Xt, K = 8)

summary(fit)

# Active basis counts are returned invisibly
s <- summary(fit)
s$active_bases
```

`toy_curves` *Toy Simulated Functional Dataset*

Description

A small simulated dataset of three functional curves used in package examples. Curves are generated from a known cubic B-spline expansion with correlated errors, making it suitable for demonstrating basis selection and recovery of true coefficients.

Usage

```
toy_curves
```

Format

A list with the following elements:

`y` Named list of 3 numeric vectors of length 50, one per curve.

`Xt` Numeric vector of 50 equally spaced time points on $[0, 1]$.

`true_coef` Numeric vector of length 8. True basis coefficients: `c(1.5, 0, -1, 0.8, 0, -0.5, 1.2, -0.9)`.

`K` Integer. Number of basis functions used (`8`).

`m` Integer. Number of curves (`3`).

`sigma` Numeric. True noise standard deviation (`0.1`).

`w` Numeric. True correlation decay parameter (`6`).

Details

Each curve is generated as:

$$y_i(t) = \sum_{k=1}^{8} \xi_{ki} B_k(t) + \varepsilon_i(t)$$

where $(\boldsymbol{\xi}_i) = (1.5, 0, -1, 0.8, 0, -0.5, 1.2, -0.9)$ for all i, and $\varepsilon_i \sim \text{GP}(0, \sigma^2 \Psi(w))$ with $\sigma = 0.1$ and $w = 6$ (correlation function of an Ornstein-Uhlenbeck (OU) process). Basis functions 2 and 5 have zero coefficients, providing a ground truth for evaluating basis selection.

Source

Generated via `data-raw/generate_toy_curves.R`.

Examples

```
data(toy_curves)
str(toy_curves)

# Plot the three raw curves
plot(toy_curves$Xt, toy_curves$y[[1]], type = "l",
     ylab = "y", xlab = "t", main = "Toy curves")
lines(toy_curves$Xt, toy_curves$y[[2]], col = "blue")
lines(toy_curves$Xt, toy_curves$y[[3]], col = "red")
```

---

tune_vem_by_gcv *Tune Basis Complexity via GCV*

Description

Fits `vem_smooth` across a grid of candidate basis sizes `K_grid` and selects the best K using GCV scores from `gcv_vem`. Called internally by `vem_fit` when a vector of K values is supplied; not typically called directly.

Two selection modes are supported: `"mean"` selects the single K minimizing the mean GCV across all curves; `"per_curve"` selects the K that minimizes the GCV criterion for each individual curve, producing a composite fit.

Usage

```
tune_vem_by_gcv(
  y,
  Xt,
  K_grid,
  build_B,
  initial_values_fn,
  threshold = 0.5,
  mode = c("mean", "per_curve"),
  ...
)
```

Arguments

`y`	List of curves.
`Xt`	Numeric vector of time points.
`K_grid`	Integer vector of candidate K values.
`build_B`	Function with signature `function(K, Xt, y)` that returns a list of $n \times K$ basis matrices.

`initial_values_fn`
: Function with signature `function(K, m)` that returns an `initial_values` list for `vem_smooth`.

`threshold`
: Posterior inclusion probability (PIP) threshold passed to `gcv_vem`. Default `0.5`.

`mode`
: Character. `"mean"` for a single global K; `"per_curve"` for curve-specific K. Default `"mean"`.

`...`
: Additional arguments passed to `vem_smooth`.

Value

A list with:

`fits` Named list of fitted `vem_smooth` objects, one per candidate K.

`gcv_matrix` Numeric matrix ($m \times$ `length(K_grid)`) of per-curve GCV scores.

`best_K_mean` Integer. Best K by mean GCV.

`best_K_per_curve` Integer vector of length m. Best K for each curve.

References

da Cruz, A. C., de Souza, C. P. E., & Sousa, P. H. T. O. (2024). Fast Bayesian basis selection for functional data representation with correlated errors. *arXiv:2405.20758*. https://arxiv.org/abs/2405.20758

See Also

`vem_fit`, `gcv_vem`

---

vem_fit *Fit a VEM Smooth Model*

Description

Fits one or more functional curves using Bayesian basis function selection via the Variational EM algorithm, with an Ornstein-Uhlenbeck within-curve correlation structure. Internally calls `vem_smooth` to run the VEM algorithm.

If a single value is provided for K, the model is fitted using that fixed number of basis functions. If a vector of candidate values is supplied, the function `tune_vem_by_gcv` is called to automatically select the optimal K based on the GCV criterion. The resulting fitted object provides methods for plot, predict, coef, and summary via the corresponding S3 methods `plot.vem_fit`, `predict.vem_fit`, `coef.vem_fit`, and `summary.vem_fit`.

Usage

```
vem_fit(
  y,
  Xt,
  K = NULL,
  basis_type = c("cubic_bspline", "fourier"),
  selection_metric = c("mean", "per_curve"),
  threshold = 0.5,
  center = FALSE,
  scale = FALSE,
  period = NULL,
  initial_values_fn = NULL,
  lambda_1 = NULL,
  lambda_2 = NULL,
  delta_1 = NULL,
  delta_2 = NULL,
  ...
)
```

Arguments

y	Named list of numeric vectors, one per curve.
Xt	Numeric vector of time points, common across all curves.
K	Integer or integer vector of candidate basis sizes. If a single value, fits directly at that K. If a vector, selects best K via GCV. If NULL, defaults to c(10, 15, 20, 30) for B-splines and Fourier.
basis_type	Character. One of "cubic_bspline" (default), or "fourier".
selection_metric	Character. "mean" selects a single global K minimizing mean GCV across curves. "per_curve" selects the best K independently for each curve, returning a composite fit. Only relevant when K is a vector. Default "mean".
threshold	Numeric in (0,1). Posterior inclusion probability (PIP) threshold for active basis functions in GCV calculation. Default 0.5.
center	Logical. If TRUE, subtract each curve's mean before fitting. If TRUE, the function automatically centralizes the curves before model fitting. Default FALSE.
scale	Logical. If TRUE, the function automatically standardizes the curves before model fitting, by dividing each curve by its standard deviation. Predictions are automatically back-transformed. Default FALSE.
period	Numeric. Period for Fourier bases. Defaults to the domain range diff(range(Xt)) if NULL.
initial_values_fn	Function with signature function(K, m) returning an initial_values list for vem_smooth. If NULL, an empirical initialization based on a dense regression spline fit is used.
lambda_1, lambda_2	Positive scalars. Inverse-Gamma prior hyperparameters for τ^2. Defaults: lambda_1 = 0.001, lambda_2 = 0.001.

`delta_1`, `delta_2`
: Positive scalars. Inverse-Gamma prior hyperparameters for σ^2. Defaults: `delta_1` = `10`, `delta_2` = `0.09`.

`...`
: Additional arguments passed to `vem_smooth`, such as `maxIter`, `convergence_threshold`, and `mu_ki`.

Value

An object of class `"vem_fit"` containing:

`model` The fitted `vem_smooth` object (global fit), or a named list of per-curve model objects (composite fit).

`selected_K` Integer vector of length `m`. The `K` used for each curve.

`best_K` The single selected `K` (global fit), or a vector (composite fit).

`tuning` Output of `tune_vem_by_gcv`, including the full GCV matrix and all candidate fits. `NULL` if a single `K` was supplied.

`scaling_params` List with `means` and `sds` used for standardization. Used by `predict.vem_fit` and `plot.vem_fit` to back-transform predictions.

`data_orig` The input curves in their original scales.

`basis_type`, `is_composite`, `Xt`, `call` Metadata stored for use by S3 methods.

References

da Cruz, A. C., de Souza, C. P. E., & Sousa, P. H. T. O. (2024). Fast Bayesian basis selection for functional data representation with correlated errors. *arXiv:2405.20758*. https://arxiv.org/abs/2405.20758

See Also

`vem_smooth`, `plot.vem_fit`, `predict.vem_fit`, `coef.vem_fit`, `summary.vem_fit`

Examples

```
data(toy_curves)

fit <- vem_fit(
  y    = toy_curves$y,
  Xt   = toy_curves$Xt,
  K    = 8
)

summary(fit)
plot(fit, curve_idx = 1)
coef(fit)
predict(fit)

# GCV tuning over a grid of K values
fit_gcv <- vem_fit(
  y    = toy_curves$y,
```

```
  Xt   = toy_curves$Xt,
  K    = c(6, 8, 10)
)
fit_gcv$best_K
```

---

vem_smooth *Variational EM Algorithm for Bayesian Basis Function Selection*

Description

Fits m functional curves simultaneously via Bayesian basis function selection with an Ornstein-Uhlenbeck within-curve correlation structure. This function is called internally by `vem_fit` and only runs the VEM algorithm itself, without performing basis construction, standardization, or GCV tuning. Most users should call `vem_fit` instead, which handles those steps automatically.

Usage

```
vem_smooth(
  y,
  B,
  Xt = Xt,
  m = length(y),
  K = K,
  mu_ki = 0.5,
  lambda_1 = 1e-10,
  lambda_2 = 1e-10,
  delta_1 = 1e-10,
  delta_2 = 1e-10,
  maxIter = 1000,
  initial_values,
  convergence_threshold = 0.01,
  lower_opt = 0.1
)
```

Arguments

`y`	List of length m of numeric vectors (observed curves, possibly standardized).
`B`	List of length m of $n_i \times K$ basis matrices, typically from `getbasismatrix`.
`Xt`	Numeric vector of n evaluation points, common across curves.
`m`	Integer. Number of curves. Defaults to `length(y)`.
`K`	Integer. Number of basis functions.
`mu_ki`	Numeric scalar in $(0, 1)$. Beta prior hyperparameter for inclusion probabilities. Default `0.5`.
`lambda_1, lambda_2`	Positive scalars. Inverse-Gamma prior hyperparameters for τ^2. Default 10^{-10}.

`delta_1`, `delta_2`
: Positive scalars. Inverse-Gamma prior hyperparameters for σ^2. Default 10^{-10}.

`maxIter`
: Integer. Maximum VEM iterations. Default `1000`.

`initial_values`
: Named list with elements `p` (inclusion probabilities, length mK), `delta2`, `lambda2`, and `w`.

`convergence_threshold`
: Positive scalar. Absolute ELBO tolerance for convergence. Default `0.01`.

`lower_opt`
: Positive scalar. Lower bound for w in L-BFGS-B. Default `0.1`.

Details

The algorithm alternates between an E-step — sequential coordinate ascent variational inference (CAVI) updates for $q(\beta_i)$, $q(\sigma^2)$, $q(\tau^2)$, $q(Z_{ki})$, and $q(\theta_{ki})$ — and an M-step that maximizes the ELBO with respect to the correlation decay parameter w via L-BFGS-B with an analytic gradient. Convergence is declared when the absolute ELBO change between iterations falls below `convergence_threshold`.

For hyperparameter initialization, set `delta_1` and `delta_2` such that `delta_2 / (delta_1 - 1)` is a rough estimate of the noise variance, and initialize `w` consistent with the expected correlation strength in the data.

Value

A named list containing:

`mu_beta` Posterior means $\mu_{\beta_{ki}}$ (length mK).

`Sigma_beta` Posterior covariance array ($K \times K \times m$).

`prob` Posterior inclusion probabilities p_{ki} (length mK). Basis k is active for curve i when $p_{ki} > 0.5$.

`delta1`, `delta2` Final $q(\sigma^2)$ parameters.

`lambda1`, `lambda2` Final $q(\tau^2)$ parameters.

`w` Estimated correlation decay parameter (range-normalized scale).

`cor_mat` The $n \times n$ Ornstein-Uhlenbeck correlation matrix Ψ evaluated at the final estimated decay parameter $\hat{w}$, as returned by `computePsiMatrix`.

`elbo_values` ELBO trajectory across iterations.

`converged` Logical. Whether the convergence criterion was met.

`n_iterations` Number of iterations run.

References

da Cruz, A. C., de Souza, C. P. E., & Sousa, P. H. T. O. (2024). Fast Bayesian basis selection for functional data representation with correlated errors. *arXiv:2405.20758*. https://arxiv.org/abs/2405.20758

See Also

`vem_fit`, `plot.vem_fit`, `predict.vem_fit`, `coef.vem_fit`

Index