Package 'VCMoE'

July 15, 2026

Title Varying-Coefficient Mixture-of-Experts Models

Version 0.2.0

Description Fits Gaussian, Binomial, and Negative-Binomial varying-coefficient mixture-of-experts models with local-linear estimation, explicit label alignment, bandwidth selection, diagnostics, bootstrap inference, analytic-style confidence bands, coefficient-specific analytic generalized likelihood-ratio test (GLRT) diagnostics with optional bootstrap calibration, and local-grid or joint-path expectation-maximization fitting engines.

License MIT + file LICENSE

URL https://qc-zhao.github.io/VCMoE/, https://github.com/qc-zhao/VCMoE

BugReports https://github.com/qc-zhao/VCMoE/issues

Encoding UTF-8

RoxygenNote 7.3.3

Imports ggplot2, stats, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Qicheng Zhao [aut, cre],
Celia M.T. Greenwood [aut],
Qihuang Zhang [aut]

Maintainer Qicheng Zhao <790471666@qq.com>

Repository CRAN

Date/Publication 2026-07-15 08:10:22 UTC

Contents

`coef.vcmoe` *Extract VCMoE coefficients*

Description

Extracts expert coefficients, gating coefficients, Gaussian variance intercepts, Gaussian log-sigma local-linear slopes, Negative-Binomial theta, or all fitted coefficient blocks from a VCMoE fit.

Usage

```
## S3 method for class 'vcmoe'
coef(object, type = c("all", "expert", "gating", "sigma", "sigma_slope", "theta"), ...)
```

Arguments

`object`	A `vcmoe` object.
`type`	Coefficient block to return.
`...`	Unused.

Details

For Gaussian fits, `coef(fit, "sigma")` returns the component-specific standard deviation function at each `u_grid` point, and `coef(fit, "sigma_slope")` returns the scaled local-linear slope of `log(sigma)` on the `(u - u0) / h` basis.

For Binomial fits, expert coefficients are on the logit success-probability scale and `coef(fit, "sigma")` returns `NULL`. For Negative-Binomial fits, expert coefficients are on the log mean count scale, `coef(fit, "theta")` returns the component-specific size parameter, and `coef(fit, "sigma")` returns `NULL`.

Value

A list or array of fitted coefficient functions.

`confint.vcmoe_bootstrap` *Bootstrap confidence intervals for VCMoE coefficients*

Description

Summarizes pointwise or simultaneous bootstrap intervals for expert or gating coefficient functions.

Usage

```
## S3 method for class 'vcmoe_bootstrap'
confint(
  object,
  parm = c("expert", "gating"),
  level = 0.95,
  type = c("pointwise", "simultaneous"),
  ...
)
```

Arguments

`object`	A `vcmoe_bootstrap` object.
`parm`	Coefficient set to summarize: `"expert"`, `"gating"`, or both.
`level`	Confidence level.
`type`	Interval type. `"pointwise"` uses percentile bootstrap intervals. `"simultaneous"` uses a max standardized bootstrap-deviation band over `u` for each component and term.
`...`	Unused.

Details

Pointwise intervals are percentile intervals at each grid point. Simultaneous bands compute bootstrap standard errors and use the empirical quantile of the maximum standardized absolute deviation over the `u_grid`. Near-zero standard errors are floored internally to avoid division by zero.

Value

A tidy data frame with columns `coefficient_set`, `term`, `component`, `u`, `estimate`, `se`, `lower`, `upper`, `type`, `level`, and `n_successful`.

plot_coefficients *Plot fitted coefficient functions*

Description

Plots fitted expert or gating coefficient functions over the VCMoE grid.

Usage

```
plot_coefficients(object, type = c("expert", "gating"))
```

Arguments

object	A vcmoe object.
type	"expert" or "gating".

Value

A ggplot object.

plot_diagnostics *Plot VCMoE fit diagnostics*

Description

Plots convergence, posterior entropy, component proportions, effective local sample size, and label ambiguity flags over the coefficient grid.

Usage

```
plot_diagnostics(object)
```

Arguments

object	A vcmoe object.

Details

This plot is intended as a first real-data sanity check before interpreting coefficient functions. Ambiguity or non-convergence at many grid points should be treated as evidence that the fitted component labels or coefficient paths need closer review.

Value

A ggplot object.

plot_inference *Plot bootstrap inference intervals*

Description

Plots fitted coefficient functions with bootstrap pointwise intervals or simultaneous bands.

Usage

```
plot_inference(
  object,
  coefficient_set = "expert",
  type = c("pointwise", "simultaneous"),
  level = 0.95
)
```

Arguments

object	A vcmoe_bootstrap object.
coefficient_set	Coefficient set to plot: "expert" or "gating".
type	Interval type passed to confint().
level	Confidence level.

Value

A ggplot object.

plot_posterior *Plot fitted posterior summaries*

Description

Plots mean posterior probabilities for each component over the coefficient grid.

Usage

```
plot_posterior(object)
```

Arguments

object	A vcmoe object.

Value

A ggplot object.

predict.vcmoe *Predict from a VCMoE fit*

Description

Returns fitted means, component-specific means, posterior probabilities, or gating probabilities.

Usage

```
## S3 method for class 'vcmoe'
predict(object, newdata = NULL, u = NULL,
  type = c("mean", "posterior", "component", "prior"), ...)
```

Arguments

object	A vcmoe object.
newdata	Optional data frame.
u	Optional index values for newdata.
type	Prediction type.
...	Unused.

Details

For Gaussian fits, type = "component" returns component-specific means and type = "mean" returns the posterior-weighted fitted mean. For Binomial fits, type = "component" returns component-specific success probabilities and type = "mean" returns the marginal success probability. For Negative-Binomial fits, type = "component" returns component-specific mean counts and type = "mean" returns the marginal mean count.

Value

A vector or matrix depending on type.

simulate_vcmoe_binomial *Simulate Binomial VCMoE data*

Description

Generates Binomial VCMoE simulations for Bernoulli and grouped-count examples and tests.

Usage

```
simulate_vcmoe_binomial(n = 300L, k = 2L, seed = NULL,
  separation = 1, u = NULL, scenario = "well_separated", trials = 1L)
```

Arguments

n	Number of observations.
k	Number of components. Values 2 through 10 are supported.
seed	Optional random seed.
separation	Controls expert separation.
u	Optional numeric vector of index values.
scenario	Simulation scenario: "well_separated", "moderate", "near_overlap", "crossing", or "imbalanced_gating".
trials	Binomial trial counts. Use 1 for Bernoulli data, or a positive integer scalar/vector for grouped Binomial data.

Value

A list with data and truth. Expert truth is on the logit scale. The truth entry includes component coefficients, gating logits, component probabilities, component-specific success probabilities, sampled class labels, and success/failure counts.

---

simulate_vcmoe_gaussian

Simulate Gaussian VCMoE data

Description

Generates a small Gaussian no-offset VCMoE simulation for tutorials and tests.

Usage

```
simulate_vcmoe_gaussian(n = 300L, k = 2L, seed = NULL,
  separation = 1, u = NULL, scenario = "well_separated")
```

Arguments

n	Number of observations.
k	Number of components. Values 2 through 10 are supported.
seed	Optional random seed.
separation	Controls expert separation.
u	Optional numeric vector of index values.
scenario	Simulation scenario: "well_separated", "moderate", "near_overlap", "crossing", or "imbalanced_gating".

Value

A list with data and truth. The truth entry includes component coefficients, gating logits, probabilities, means, standard deviations, and sampled class labels.

simulate_vcmoe_negbin *Simulate Negative-Binomial VCMoE count data*

Description

Generates Negative-Binomial VCMoE simulations for gene-expression count examples and tests.

Usage

```
simulate_vcmoe_negbin(n = 300L, k = 2L, seed = NULL,
  separation = 1, u = NULL, scenario = "well_separated",
  size_factor = NULL, mean_count = 5)
```

Arguments

n	Number of observations.
k	Number of components. Values 2 through 10 are supported.
seed	Optional random seed.
separation	Controls expert separation.
u	Optional numeric vector of index values.
scenario	Simulation scenario: "well_separated", "moderate", "near_overlap", "crossing", or "imbalanced_gating".
size_factor	Optional positive size factors. If NULL, log-normal size factors are generated.
mean_count	Baseline count scale.

Value

A list with data and truth. Expert truth is on the log mean count scale. The data include size_factor and log_size_factor for use with offset(log_size_factor).

vcmoe_bootstrap *Parametric bootstrap inference for a VCMoE fit*

Description

Parametric bootstrap inference for a VCMoE fit

Usage

```
vcmoe_bootstrap(
  fit,
  data,
  u = NULL,
  B = 200L,
  coefficient_set = c("expert", "gating"),
  seed = NULL,
  control = list(),
  min_successful = max(20L, ceiling(0.5 * B)),
  keep_fits = FALSE,
  verbose = FALSE
)
```

Arguments

`fit`	A `vcmoe` fit with `k = 2:10`.
`data`	Original data frame used to fit `fit`.
`u`	Optional original `u` values or column name.
`B`	Number of bootstrap replicates.
`coefficient_set`	Coefficient sets to store.
`seed`	Optional random seed.
`control`	Control overrides for bootstrap refits.
`min_successful`	Minimum successful replicates for reliable inference.
`keep_fits`	Whether to store successful bootstrap fit objects.
`verbose`	Whether to message progress.

Details

Bootstrap refits preserve the reference fitting engine. A joint-path reference is therefore refitted with `vcmoe_fit(..., engine = "joint_path_em")` rather than silently falling back to local-grid EM.

Value

An object of class `vcmoe_bootstrap`.

vcmoe_confband *Analytic-style confidence bands for a VCMoE fit*

Description

Analytic-style confidence bands for a VCMoE fit

Usage

```
vcmoe_confband(
  fit,
  data = NULL,
  level = 0.95,
  type = c("pointwise", "simultaneous"),
  coefficient_set = c("expert", "gating", "sigma", "theta"),
  strict = TRUE,
  control = list()
)
```

Arguments

fit	A vcmoe fit with k = 2:10.
data	Optional original data frame. The current implementation uses the data stored in fit$fitted; refit with keep_data = TRUE if needed.
level	Confidence level.
type	Interval columns to expose as lower and upper.
coefficient_set	Coefficient blocks to return.
strict	Whether weak local fits should return blocked intervals.
control	Optional development inference controls. HC0 is the only active covariance adjustment.

Details

For engine = "joint_path_em", the covariance follows the JASA observed local-likelihood asymptotic sandwich plug-in. It does not include shared-path, label-selection, or finite-grid cross-grid responsibility uncertainty. Joint-path convergence and the returned score-imbalance diagnostics should therefore be inspected. The returned metadata identifies the covariance target, estimator/covariance match, omitted uncertainty, and coverage-theory scope; no bias or boundary correction is applied.

Value

A vcmoe_confband object with interval and diagnostic data frames.

vcmoe_diagnostics *Summarize VCMoE fit diagnostics*

Description

Returns a compact diagnostic table for reviewing whether a fitted VCMoE model is reliable enough to interpret.

Usage

```
vcmoe_diagnostics(object)
```

Arguments

object	A vcmoe object.

Details

The table includes convergence status, iterations, local log-likelihood, local-weighted posterior entropy, label ambiguity flags, alignment margin, effective local sample size, local-weighted component posterior proportions, and Binomial expert optimizer diagnostics when available.

Posterior entropy and component proportions use the same local kernel weights as the fitted grid point when the fit retains training data. If the fit was created with `control$keep_data = FALSE`, component proportions fall back to unweighted posterior means and effective local sample size is `NA`.

Value

A data frame with one row per coefficient grid point.

vcmoe_fit *Fit a varying-coefficient mixture-of-experts model*

Description

Fit a varying-coefficient mixture-of-experts model

Usage

```
vcmoe_fit(
  formula,
  data,
  u,
  k = 2L,
  family = "gaussian",
```

```
  bandwidth = NULL,
  u_grid = NULL,
  control = list(),
  label = "align",
  parameterization = "a1_epanechnikov_scaled",
  u_scale = c("unit", "none"),
  engine = c("local_grid_em", "joint_path_em"),
  progress = NULL
)
```

Arguments

formula	A formula of the form `y ~ expert_terms | gating_terms`. For grouped binomial data, use `cbind(success, failure) ~ expert_terms | gating_terms`. For Negative-Binomial count data, expert-side `offset(log_size_factor)` terms are supported.
data	A data frame.
u	Continuous index column name or numeric vector.
k	Number of mixture components. `k = 2` is the primary v0 target; `k = 3:10` are high-k candidate support and require diagnostics before being treated as stable.
family	Model family. `"gaussian"`, `"binomial"`, and `"negative-binomial"` are implemented.
bandwidth	Kernel bandwidth. If `NULL`, a Silverman-style default is used for `u`.
u_grid	Grid where coefficient functions are estimated.
control	Named list overriding EM and label-alignment settings.
label	Label strategy. `"align"` uses exact global alignment for `k <= 6` and sequential assignment with ambiguity margins for `k >= 7`; `"global"` requests exact global alignment when feasible and falls back to the same sequential assignment path for `k >= 7`; `"greedy"` keeps the older one-step alignment.
parameterization	Estimator convention. The public package uses `"a1_epanechnikov_scaled"`: Epanechnikov density weights `K((u-u0)/h)/h` and the scaled local-linear basis `(u-u0)/h`.
u_scale	How to transform `u` before fitting. The default `"unit"` maps complete-row `u` values to `[0, 1]`; `"none"` leaves `u` on the supplied scale. `bandwidth` and `u_grid` are interpreted on the transformed analysis scale.
engine	Fitting engine. The default `"local_grid_em"` preserves the original independent local-grid EM path. `"joint_path_em"` updates a shared observation-level responsibility path across grid points.
progress	Joint-path progress reporting. The default `NULL` is silent; use `TRUE` for messages or a single CSV file path for structured logging. Iteration frequency is controlled by `control$progress_every`.

Details

Rows with missing or non-finite response, covariates, or u are removed consistently before fitting, with a warning. For single-trial Bernoulli responses, the default gating ridge is strengthened to control$ridge = 1 unless the user explicitly supplies control$ridge; grouped Binomial and other families keep the global default. Joint-path traces record the sample-level nearest-grid log-likelihood as a diagnostic criterion. The label-consistent updates do not guarantee that this diagnostic is monotone; convergence is based on posterior and parameter deltas instead.

Value

An object of class vcmoe.

---

vcmoe_fit_reduced *Fit a block-constant reduced VCMoE model*

Description

Refits a VCMoE object under a block-constant coefficient constraint. A local-grid reference uses the established constrained BFGS optimizer; a joint-path reference uses a paper-inspired sample-weighted grid projection and applies the selected constraint after every M-step. This projection is not a generic constrained optimizer, and its diagnostic likelihood need not be monotone.

Usage

```
vcmoe_fit_reduced(
  fit,
  constrain = c("gating_constant", "expert_constant", "all_constant"),
  control = list()
)
```

Arguments

fit	A vcmoe fit.
constrain	Constraint to impose. "gating_constant" freezes the gating contrast functions in u; "expert_constant" freezes expert mean functions and family dispersion paths; "all_constant" freezes all fitted coefficient functions.
control	Controls for constrained null fitting. With the default strict = TRUE, a nonconverged reduced fit is rejected.

Value

A reduced object of class vcmoe with recomputed posterior and likelihood caches.

vcmoe_gating_contrasts *Report Identifiable VCMoE Gating Contrasts*

Description

Report identifiable gating contrasts.

Usage

```
vcmoe_gating_contrasts(object, baseline = NULL, scaled = FALSE)
```

Arguments

object	A vcmoe object.
baseline	Component used as the contrast baseline. By default, k = 2 uses component 2 and k > 2 uses component 1.
scaled	If TRUE, slope contrasts are converted to the scaled local-linear basis (u - u0) / h.

Details

VCMoE stores gating coefficients as centered logits, so the absolute level of all component logits is not identifiable. Interpretable gating effects are component contrasts such as component 1 versus component 2 for k = 2, or component 1 and component 2 versus component 3 for k = 3 comparisons when baseline = 3.

Value

A data frame with one row per grid point, contrast, term, and block.

vcmoe_glrt *Generalized likelihood-ratio test for VCMoE coefficient variation*

Description

Generalized likelihood-ratio test for VCMoE coefficient variation

Usage

```
vcmoe_glrt(
  fit,
  data,
  test = c("coefficient", "constant_block", "constant_all"),
  coefficient_set = c("expert", "gating", "sigma", "theta"),
  component = NULL,
  term = NULL,
  calibration = c("none", "bootstrap", "analytic_epanechnikov", "both",
    "parametric_bootstrap"),
  B = 200L,
  seed = NULL,
  control = list(),
  refit_control = list(),
  verbose = FALSE
)
```

Arguments

fit	A vcmoe fit.
data	Original data frame used to fit fit.
test	Test type. "coefficient" tests one coefficient function; "constant_block" tests all expert or gating functions jointly; "constant_all" tests all fitted coefficient functions jointly.
coefficient_set	Coefficient block for coefficient-specific or block-constant tests.
component	Component label or index for coefficient-specific tests.
term	Term name for coefficient-specific tests.
calibration	Calibration method. The default "none" returns the statistic without attaching a reference distribution. "analytic_epanechnikov" uses the Epanechnikov modified chi-square calibration; "bootstrap" uses parametric bootstrap calibration; "both" reports both. The analytic calibration is retained as an explicitly requested approximation because the implemented statistic is not identical to the manuscript criterion.
B	Number of bootstrap calibration replicates.
seed	Optional random seed.
control	Controls for constrained null optimization and diagnostics.
refit_control	Controls overriding bootstrap full-model refits.
verbose	Whether to message bootstrap progress.

Details

Local-grid fits retain the 0.1.0 constrained BFGS null optimizer. Joint-path fits use a paper-inspired sample-weighted grid-projected null: after every M-step, each constrained coefficient path is replaced by its mean weighted by the number of observations assigned to each nearest grid point, and

constrained local slopes are set to zero. Its statistic compares sample-level likelihood contributions evaluated at each observation's nearest grid point. The projected update is not a generic constrained optimizer and its diagnostic likelihood trace need not be monotone. Bootstrap calibration preserves both the full-fit engine and its matching null engine.

Value

A `vcmoe_glrt` object.

`vcmoe_parameterization` *Inspect VCMoE Parameterization Metadata*

Description

Inspect VCMoE parameterization metadata.

Usage

```
vcmoe_parameterization(object)
```

Arguments

`object`	A `vcmoe` object.

Details

The package default is `"a1_epanechnikov_scaled"`: Epanechnikov density weights `0.75 * (1 - t^2)_+ / h` with `t = (u - u0) / h`, scaled local-linear slope storage, and centered gating logits. This helper reports those conventions for reproducible model summaries.

Value

A named list describing the estimator convention used by the fit, including kernel weights, local-linear basis scale, gating-logit storage, dispersion block, label-alignment method, and optimization controls.

vcmoe_scaled_slopes *Inspect VCMoE Local-Linear Slopes On The Scaled Basis*

Description

Inspect VCMoE local-linear slopes on the scaled basis.

Usage

```
vcmoe_scaled_slopes(object, type = c("expert", "gating"), bandwidth = NULL)
```

Arguments

object	A vcmoe object.
type	Coefficient block, either "expert" or "gating".
bandwidth	Optional bandwidth recorded in the returned attributes. Defaults to the fitted bandwidth.

Details

VCMoE stores slopes on the scaled local-linear basis (u - u0) / h. This helper returns the stored scaled-basis slope block.

Value

An array with the same dimensions as the stored slope block.

vcmoe_select_bandwidth *Select a VCMoE bandwidth by K-fold cross-validation*

Description

Select a VCMoE bandwidth by K-fold cross-validation

Usage

```
vcmoe_select_bandwidth(
  formula,
  data,
  u,
  k = 2L,
  family = "gaussian",
  bandwidth_grid = NULL,
  folds = 5L,
```

```
  u_grid = NULL,
  control = list(),
  label = "align",
  parameterization = "a1_epanechnikov_scaled",
  u_scale = c("unit", "none"),
  seed = NULL,
  refit = TRUE,
  engine = c("local_grid_em", "joint_path_em")
)
```

Arguments

formula	A formula of the form y ~ expert_terms | gating_terms.
data	A data frame.
u	Continuous index column name or numeric vector.
k	Number of mixture components. Values from 2 through 10 are supported.
family	Model family. "gaussian", "binomial", and "negative-binomial" are supported.
bandwidth_grid	Candidate bandwidth values. If NULL, uses multiples of the default bandwidth.
folds	Number of random cross-validation folds.
u_grid	Grid where coefficient functions are estimated.
control	Named list passed to the selected fitting engine.
label	Label strategy passed to the selected fitting engine.
parameterization	Estimator convention passed to the selected fitting engine.
u_scale	u scaling strategy passed to the selected fitting engine.
seed	Optional random seed for fold assignment and, when control$seed is absent, deterministic CV refits.
refit	Whether to refit the final model on all data using the selected bandwidth.
engine	Fitting engine used for every cross-validation fold and the optional final refit. The default "local_grid_em" preserves the 0.1.0 behavior; "joint_path_em" uses the joint-path engine of vcmoe_fit().

Value

An object of class vcmoe_bandwidth_selection.

Index