Package: VCMoE (via r-universe)

June 20, 2026

Title Varying-Coefficient Mixture-of-Experts Models

Version 0.1.0

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, and coefficient-specific analytic GLRT diagnostics with optional bootstrap calibration.

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

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports ggplot2, stats, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Repository https://qc-zhao.r-universe.dev

Date/Publication 2026-06-20 17:49:53 UTC

RemoteUrl https://github.com/qc-zhao/vcmoe

RemoteRef HEAD

RemoteSha 8b34854676c73dd498e2c492aaafccd098488291

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

Runs parametric bootstrap inference for a fitted Gaussian, Binomial, or Negative-Binomial VCMoE model with k = 2:10.

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 fitted vcmoe object. Bootstrap inference supports k = 2:10.
data	Original data frame used to fit fit. The function resamples from data[fit$rows_used,].
u	Optional original u values or column name. If NULL, the stored u column from fit is reused when available.
B	Number of parametric bootstrap replicates.
coefficient_set	Coefficient sets to store and summarize: "expert", "gating", or both.
seed	Optional random seed.
control	Named list passed to bootstrap refits. Bandwidth is not reselected inside bootstrap v0.
min_successful	Minimum number of successful replicates expected for reliable inference. The object is returned when at least two replicates succeed, but a warning is recorded below this threshold.
keep_fits	Whether to store successful bootstrap fit objects.
verbose	Whether to print replicate progress messages.

Details

For Gaussian fits, each bootstrap data set draws a latent component from the fitted gating probabilities and then draws the response from the selected component Normal distribution. For Binomial fits, each bootstrap data set draws success counts from the selected component success probability. For Negative-Binomial fits, each bootstrap data set draws counts from the selected component mean and theta. Bernoulli and grouped cbind(success, failure) response formats are preserved for Binomial fits.

Each bootstrap replicate is refit with the same formula, family, number of components, bandwidth, u_grid, and label strategy as the reference fit. After the usual within-grid label alignment, one global component permutation matches the bootstrap coefficient paths back to the reference fit.

Ambiguous bootstrap-to-reference matches are recorded in `alignment_summary`. Exact permutation matching is used for small `k`; assignment-based matching is used when exhaustive permutation is infeasible.

Binomial expert coefficients and intervals are on the logit coefficient scale. Negative-Binomial expert coefficients and intervals are on the log mean count scale.

Value

An object of class `vcmoe_bootstrap` with fields `fit`, `replicates`, `replicate_summary`, `alignment_summary`, `settings`, `warnings`, and optionally `fits`.

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

Description

Computes HC0 analytic-style Epanechnikov path confidence bands for VCMoE fits with `k = 2:10` using the fitted Epanechnikov/scaled parameterization. High-k intervals are diagnostic-gated and should be interpreted with the returned block and Hessian diagnostics.

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.
`data`	Optional original data frame. The implementation uses `fit$fitted`; refit with `control = list(keep_data = TRUE)` if needed.
`level`	Confidence level.
`type`	Whether the convenience `lower` and `upper` columns use pointwise intervals or simultaneous bands.
`coefficient_set`	Coefficient blocks to return.
`strict`	Whether weak local fits should return blocked intervals.
`control`	Optional inference controls. HC0 is the only active covariance adjustment.

Details

The returned interval table includes pointwise and simultaneous columns, diagnostic status, block reasons, Hessian condition, effective local sample size, and SCB metadata. Binomial expert intervals are on the logit coefficient scale, Negative-Binomial expert intervals are on the log mean scale, and Negative-Binomial `theta` intervals are nuisance diagnostics.

Value

A `vcmoe_confband` object with `intervals`, `diagnostics`, and `settings`.

`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

Fits a Gaussian, Binomial, or Negative-Binomial VCMoE model by local-linear EM and aligns component labels across the coefficient-function grid.

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"))
```

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, use expert-side offset(log_size_factor) for library-size or size-factor offsets.
data	A data frame.
u	Continuous index column name or numeric vector.
k	Number of mixture components. Values 2 through 10 are accepted. High-k fits are candidate support and require diagnostics.
family	Model family. "gaussian", "binomial", and "negative-binomial" are implemented.
bandwidth	Kernel bandwidth. If NULL, a default is used.
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 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.
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.
parameterization	Estimator convention. The public package uses "a1_epanechnikov_scaled": Epanechnikov density weights 0.75 * (1 - t^2)_+ / h with t = (u - u0) / h and stores local-linear slopes on the scaled (u - u0) / h basis.

Details

Rows with missing or non-finite response, covariates, or u are removed consistently before fitting, with a warning.

By default, local EM uses Epanechnikov density kernel weights and a scaled local-linear basis. Complete-row u values are mapped to `[0, 1]` before fitting unless `u_scale = "none"`. Fitted objects store both the analysis-scale grid and the original-scale grid metadata.

For `family = "gaussian"`, the expert mean and log-standard-deviation blocks are both local-linear inside each fit. The variance block uses `log(sigma_ic) = delta_c0(u0) + delta_c1(u0) * (u_i - u0) / h`; `coef(fit, "sigma")` returns `exp(delta_c0(u0))` and `coef(fit, "sigma_slope")` returns `delta_c1(u0)`.

For `family = "binomial"`, Bernoulli responses must be 0/1 values and grouped responses must use non-negative finite whole-number success/failure counts with positive row totals. Binomial expert coefficients are on the logit success-probability scale. Binomial `k >= 3` fits are accepted as experimental stress support and should not be treated as stable inference-ready support. Binomial expert logistic M-steps use `control$binomial_ridge` with default value `1`, and `control$binomial_structured_starts = TRUE` uses structured local starts to improve single-trial Bernoulli stability. For single-trial Bernoulli responses only, the gating ridge default is also strengthened to `control$ridge = 1` unless the user explicitly supplies `control$ridge`; grouped Binomial responses keep the global default.

For `family = "negative-binomial"`, responses must be finite non-negative whole-number counts. Expert coefficients are on the log mean count scale. Use `offset(log_size_factor)` in the expert formula to account for library size or size factors. Gating-side offsets are not supported in v0. `coef(fit, "theta")` returns component-specific NB size parameters.

The default label strategy performs local EM at each grid point, then applies a post-processing dynamic-programming alignment over the full grid. Transition costs use the previous local-linear slope to predict the next coefficient value, plus gating, posterior, and, for Gaussian fits, variance consistency terms. This improves label path tracking without changing the local EM estimating equations.

For `k <= 6`, `label = "align"` uses exact derivative-aware global alignment. For `k >= 7`, `label = "align"` and `label = "global"` use sequential pairwise assignment alignment to avoid factorial permutation growth and record the best-vs-second-best assignment margin for ambiguity diagnostics.

Value

An object of class `vcmoe`.

`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 *Coefficient-specific GLRT for VCMoE coefficient variation*

Description

Fits a constrained null under the same local objective and computes a generalized likelihood-ratio statistic against the supplied VCMoE fit.

Usage

```
vcmoe_glrt(
  fit,
  data,
  test = c("coefficient", "constant_all"),
  coefficient_set = c("expert", "gating", "sigma", "theta"),
  component = NULL,
  term = NULL,
  calibration = c("analytic_epanechnikov", "bootstrap", "both", "none",
    "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_all" tests all fitted coefficient functions jointly.
coefficient_set	Coefficient block for coefficient-specific tests.
component	Component label or index for coefficient-specific tests.
term	Term name for coefficient-specific tests.
calibration	Calibration method. "analytic_epanechnikov" uses the Epanechnikov modified chi-square calibration; "bootstrap" uses parametric bootstrap calibration; "both" reports both. "parametric_bootstrap" is accepted as a backwards-compatible alias for "bootstrap".
B	Number of bootstrap calibration replicates.
seed	Optional random seed.
control	Controls for the constrained null optimizer.
refit_control	Controls overriding bootstrap full-model refits.
verbose	Whether to message progress.

Details

The primary path is coefficient-specific. The selected coefficient function is constrained to be constant in u: its local-linear slope is fixed to zero and its intercept is shared across grid points. The null is re-optimized rather than obtained by post-hoc averaging.

Analytic Epanechnikov calibration requires Epanechnikov density weights, scaled local-linear basis, and unit-scaled u. It reports lambda = ell_full - ell_null, analytic_statistic = rK * lambda, and a modified chi-square analytic_p_value. The diagnostic lrt_statistic = 2 * lambda is reported separately and is not used for the analytic p-value. Ridge penalties may be used to stabilize fitting but are excluded from the GLRT likelihood ratio.

vcmoe_glrt() supports fitted k = 2:10 models for both coefficient-specific and "constant_all" nulls. For k > 2, gating coefficient tests use identifiable baseline contrasts, e.g. component3_vs_component1. Reported p-values should be interpreted together with fit convergence, label ambiguity, component proportion, and null-optimizer diagnostics.

Value

A vcmoe_glrt object with the observed lambda, analytic statistic, null fit, optional bootstrap replicate summary, and calibrated p-value when available.

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

Selects the kernel bandwidth for a VCMoE model using random K-fold held-out predictive log-likelihood. The selected bandwidth is the candidate with the largest held-out likelihood after ranking fully successful candidates ahead of partial-failure candidates.

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
)
```

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 vcmoe_fit().
label	Label strategy passed to vcmoe_fit().
u_scale	u scaling strategy passed to vcmoe_fit(). The default selects bandwidths on the unit-scaled analysis domain.
parameterization	Estimator convention passed to vcmoe_fit() for each CV fold and the optional final refit.
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.

Details

The default candidate grid is the current Silverman-style default bandwidth multiplied by c(0.5, 0.75, 1, 1.25, 1.5, 2). Fold assignment is made only among complete rows that vcmoe_fit() would keep.

For Gaussian models, validation scoring uses log sum_c pi_c Normal(y | mu_c, sigma_c). For Binomial models, scoring uses log sum_c pi_c Binomial(success | trials, p_c). Binomial Bernoulli and grouped cbind(success, failure) responses are supported. For Negative-Binomial models, scoring uses log sum_c pi_c NB(y | mu_c, theta_c).

Bandwidth selection supports k = 2:10 for Gaussian, Binomial, and Negative-Binomial models. High-k candidates use the same held-out predictive likelihood scoring and should be interpreted together with the returned fit diagnostics. The selected object records the fitting parameterization and u scaling strategy in settings$parameterization and settings$u_scale.

Value

An object of class vcmoe_bandwidth_selection with fields best_bandwidth, cv_summary, cv_details, cv_folds, fit, and settings.

Index