Package: mixqrgate (via r-universe)

October 15, 2026

Type Package

Title Location-Varying Gating for Mixtures of Quantile Regressions

Version 0.1.2

Description Extends finite mixtures of quantile regressions (the 'mixqr' package) with a concomitant-covariate, quantile-indexed mixing gate. The mixing probabilities follow a multinomial-logit model whose coefficients may depend on a separate set of gating covariates and on the quantile level, so that latent-class membership can change across the conditional distribution. This turns the location-varying mixing idea of Furno (2025) into a likelihood and EM object with proper standard errors on the gate. Estimation reuses the 'mixqr' component and error-density machinery; the gate is fit by a weighted multinomial-logistic M-step and reduces exactly to a constant gate when the gating model is an intercept only.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1)

Imports mixqr (>= 0.1.1), stats, graphics

Suggests nnet, ggplot2, testthat (>= 3.0.0), knitr, rmarkdown

RoxygenNote 7.3.3

Roxygen list(markdown = TRUE)

Config/testthat/edition 3

Config/Needs/website pkgdown

VignetteBuilder knitr

URL https://github.com/kvenkita/mixqrgate

BugReports https://github.com/kvenkita/mixqrgate/issues

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

Date/Publication 2026-07-07 18:30:39 UTC

RemoteUrl https://github.com/kvenkita/mixqrgate

RemoteRef HEAD

RemoteSha 59166399d4ccf09c14867c5854b0995dafece4ba

Contents

confint.mixqrgate 2
mixqrgate 3
mixqrgate_control 4
plot.mixqrgate 5
predict.mixqrgate 6
sim_gate2 6
summary.mixqrgate 7

Index **8**

`confint.mixqrgate` *Confidence intervals for the gate coefficients*

Description

Confidence intervals for the gate coefficients

Usage

```
## S3 method for class 'mixqrgate'
confint(object, parm, level = 0.95, tau = NULL, ...)
```

Arguments

`object`	A "`mixqrgate`" object.
`parm`	Ignored; intervals are returned for all gate coefficients.
`level`	Confidence level.
`tau`	Quantile level at which to report the (location-varying) gate; defaults to the first grid point.
`...`	Unused.

Value

A matrix of lower/upper limits for the gate coefficients.

`mixqrgate` *Fit a gated mixture of quantile regressions*

Description

Fits a finite mixture of tau-quantile regressions whose mixing probabilities follow a multinomial-logit gate on concomitant covariates, optionally indexed by the quantile level. The gate turns the location-varying mixing of Furno (2025) into a likelihood/EM object with standard errors.

Usage

```
mixqrgate(
  formula,
  data,
  gating = ~1,
  G = 2L,
  tau = 0.5,
  vary_gating = c("none", "discrete"),
  method = c("ald", "kde"),
  variance = c("sandwich", "louis", "stochEM"),
  gate_B = 200L,
  control = mixqrgate_control()
)
```

Arguments

`formula`	Component model, `y ~ x1 + x2` (intercept implied).
`data`	A data frame.
`gating`	One-sided concomitant gate formula in the gating covariates; `~1` (default) reproduces a constant gate (and the plain `mixqr` fit).
`G`	Number of components. Default `2`.
`tau`	Quantile level(s) in (0, 1). A vector triggers a location-varying (per-tau) gate.
`vary_gating`	`"none"` (gate depends on covariates, not tau) or `"discrete"` (a separate gate per tau-grid point; Furno mode).
`method`	`"ald"` (parametric asymmetric-Laplace, genuine likelihood) or `"kde"` (Wu & Yao nonparametric error densities).
`variance`	Gate standard errors. `"sandwich"` (default) is the observed-information sandwich *conditional on the fitted class memberships*, so it under-covers when components overlap. `"louis"` applies Louis's (1982) identity to the gate block – an analytic, classification-aware covariance (complete information minus the missing information from the latent labels), the recommended choice for inference. `"stochEM"` is a multiple-imputation alternative (slower; a useful cross-check).
`gate_B`	Number of imputations for `variance = "stochEM"`. Default `200`.
`control`	A `mixqrgate_control()` list.

Value

An object of class "mixqrgate".

References

Furno, M. (2025). Finite Mixture at Quantiles and Expectiles. *JRFM* 18(4), 177.

Examples

```
set.seed(1)
d <- sim_gate2(n = 300)
fit <- mixqrgate(y ~ x, data = d, gating = ~ z, G = 2, tau = 0.5)
fit
```

---

mixqrgate_control *Control parameters for* mixqrgate()

Description

Control parameters for mixqrgate()

Usage

```
mixqrgate_control(
  nstart = 10L,
  maxit = 500L,
  tol = 1e-06,
  gate_ridge = 0.001,
  gate_maxit = 50L,
  gate_tol = 1e-08,
  label_order = "slope",
  bandwidth = NULL,
  kde_grid = 512L,
  trace = FALSE,
  seed = NULL
)
```

Arguments

nstart	Number of multi-start initialisations (the mixture likelihood is multimodal). Default 10.
maxit	Maximum EM iterations. Default 500.
tol	Convergence tolerance on the relative parameter change. Default 1e-6.
gate_ridge	Ridge penalty on the gate coefficients (stabilises the multinomial-logit M-step under separation / small samples). Default 1e-3.

`gate_maxit`, `gate_tol`
: Inner Newton iterations / tolerance for the gate M-step. Defaults `50` and `1e-8`.

`label_order`
: Component ordering for label switching: `"slope"` (default, ascending first slope) or `"intercept"`.

`bandwidth`
: Optional KDE bandwidth override (`method = "kde"`).

`kde_grid`
: KDE grid size (`method = "kde"`). Default `512`.

`trace`
: Logical; print EM progress. Default `FALSE`.

`seed`
: Optional integer RNG seed.

Value

A list of class `"mixqrgate_control"`.

`plot.mixqrgate` *Plot a gated mixture fit*

Description

`which = "gating"` draws the class-average gate probability against the quantile level – the location-varying mixing picture that is Furno's headline finding. `which = "fit"` draws the data coloured by class with the component quantile lines at the first grid point.

Usage

```
## S3 method for class 'mixqrgate'
plot(x, which = c("gating", "fit"), ...)
```

Arguments

`x`
: A `"mixqrgate"` object.

`which`
: `"gating"` (default) or `"fit"`.

`...`
: Passed to `graphics::plot()`.

Value

Invisibly `x`.

predict.mixqrgate *Predict gate probabilities or class membership*

Description

Predict gate probabilities or class membership

Usage

```
## S3 method for class 'mixqrgate'
predict(object, newdata = NULL, type = c("prob", "class"), tau = NULL, ...)
```

Arguments

object	A "mixqrgate" object.
newdata	Optional data frame (must contain the gating covariates).
type	"prob" (gate probabilities) or "class" (most likely class).
tau	Quantile level at which to evaluate the (location-varying) gate; defaults to the first grid point.
...	Unused.

Value

A matrix of gate probabilities (type = "prob") or an integer vector of class labels (type = "class").

sim_gate2 *Simulate a two-component mixture with a concomitant gate*

Description

Membership depends on a gating covariate z through a logit Pr(class 2 | z) = plogis(gamma[1] + gamma[2] z); the two components are quantile regressions of y on x with distinct slopes. Errors are median-zero.

Usage

```
sim_gate2(
  n = 400L,
  gamma = c(0, 1.5),
  b1 = c(2, -3),
  b2 = c(-2, 3),
  sigma = c(1, 1.5),
  loc_vary = 0,
  het = FALSE
)
```

Arguments

n	Sample size.
gamma	Length-2 gate coefficients (intercept, slope on z).
b1, b2	Length-2 (intercept, slope) for components 1 and 2.
sigma	Length-2 error scales.
loc_vary	Strength of the location-varying gate (0 = membership independent of the quantile rank; larger = stronger upper-tail tilt toward class 2).
het	If TRUE, component-2 spread grows with x (heteroscedastic).

Details

With loc_vary > 0 the gate becomes **genuinely location-varying** in the sense of Furno (2025): membership also depends on the latent quantile rank, so class 2 is over-represented in the upper tail and the class composition – hence the fitted gate – shifts across the quantile level. With het = TRUE the second component's spread grows with x.

Value

A data frame with y, x, z, and the true class.

References

Furno, M. (2025). Finite Mixture at Quantiles and Expectiles. *JRFM* 18(4), 177.

---

summary.mixqrgate *Summarize a gated mixture fit*

Description

Reports, per quantile level, the component coefficients and the **gate coefficients with standard errors, z- and p-values** – the inference on how membership depends on the gating covariates (and, across the grid, on tau).

Usage

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

Arguments

object	A "mixqrgate" object.
...	Unused.

Value

An object of class "summary.mixqrgate".

Index

confint.mixqrgate, 2

graphics::plot(), *5*

mixqrgate, 3
mixqrgate(), *4*
mixqrgate_control, 4
mixqrgate_control(), *3*

plot.mixqrgate, 5
predict.mixqrgate, 6

sim_gate2, 6
summary.mixqrgate, 7