Package 'ModalNet'

November 13, 2026

Type Package

Title Regularized Parametric Modal Regression: Lasso, Elastic Net and Ridge

Version 0.3.1

Description Variable selection and estimation for parametric modal regression with continuous positive responses (Gamma, Weibull and Lognormal families) under Lasso, elastic-net, ridge and adaptive-Lasso penalties. The conditional mode is linked to a linear predictor through a logarithmic link using the mode reparameterization of Galarza and Lachos (2026) <doi:10.48550/arXiv.2603.07099>. The penalized log-likelihood is maximized with a proximal Newton algorithm (iteratively reweighted penalized least squares): at each iteration the modal log-likelihood is replaced by a local quadratic surrogate in the linear predictor, which yields a pseudo-response and a working weight in closed form, and the resulting weighted penalized least-squares subproblem is solved by the coordinate-descent solver of 'glmnet' (Friedman, Hastie and Tibshirani, 2010, <doi:10.18637/jss.v033.i01>). The regularization parameter is chosen along a warm-started path by the extended Bayesian information criterion of Chen and Chen (2008) <doi:10.1093/biomet/asn034>.

License GPL-3

Encoding UTF-8

Depends R (>= 4.1.0)

Imports glmnet, stats, graphics

Suggests survival

LazyData true

URL https://github.com/chedgala/ModalNet

BugReports https://github.com/chedgala/ModalNet/issues

RoxygenNote 7.3.3

NeedsCompilation no

Author Christian E. Galarza [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-4818-6006>, affiliation: Escuela Superior Politécnica del Litoral, Guayaquil, Ecuador),

Víctor H. Lachos [aut] (ORCID: <https://orcid.org/0000-0002-7239-2459>,
affiliation: University of Connecticut, Storrs, CT, USA),
Dashun Liu [aut] (affiliation: University of Connecticut, Storrs, CT,
USA)

Maintainer Christian E. Galarza <chedgala@espol.edu.ec>

Repository CRAN

Date/Publication 2026-11-13 13:50:02 UTC

Contents

generate_modal_data 2
ModalNet-methods 3
modal_net 4
plot.ModalNet 7
predict.ModalNet 8
riboflavinV100 8

Index **10**

`generate_modal_data` *Simulate data from a modal regression model*

Description

Generates positive responses whose conditional mode is $M_i = \exp(\gamma_0 + x_i^\top \gamma)$ under the Gamma, Weibull or Lognormal mode reparameterization used by `modal_net`.

Usage

```
generate_modal_data(X, gamma, phi, family = c("gamma", "weibull", "lognormal"))
```

Arguments

`X`	covariate matrix without intercept.
`gamma`	coefficient vector of length `ncol(X) + 1` (intercept first).
`phi`	dispersion parameter.
`family`	one of `"gamma"`, `"weibull"`, `"lognormal"`.

Value

A list with `y`, `mode` and `mean`.

Examples

```
X <- matrix(rnorm(100 * 5), 100, 5)
d <- generate_modal_data(X, c(1, 0.5, -0.5, 0, 0, 0), phi = 0.3, family = "weibull")
str(d)
```

ModalNet-methods *Methods for ModalNet fits*

Description

Print, summarize and extract the components of a model fitted by `modal_net`.

Usage

```
## S3 method for class 'ModalNet'
print(x, digits = 4, ...)

## S3 method for class 'ModalNet'
summary(object, digits = 4, ...)

## S3 method for class 'summary.ModalNet'
print(x, digits = 4, ...)

## S3 method for class 'ModalNet'
coef(object, s = NULL, ...)

## S3 method for class 'ModalNet'
fitted(object, ...)

## S3 method for class 'ModalNet'
logLik(object, ...)
```

Arguments

`x`, `object`	a `"ModalNet"` object (`x` is a `"summary.ModalNet"` object in `print.summary.ModalNet`).
`digits`	number of decimal places to print.
`...`	unused.
`s`	optional penalty value; `coef()` then returns the coefficients of the path at the grid value closest to `s`.

Value

`print.ModalNet` and `print.summary.ModalNet` return their argument invisibly. `summary.ModalNet` returns an object of class `"summary.ModalNet"`, a list with the coefficient table of the selected covariates (estimates on the log-mode scale and their exponentials), the dispersion, the selected penalty, the log-likelihood, EBIC and degrees of freedom, the names of the selected covariates, the mode-mean constant $c(\hat{\phi})$ with the implied mean-scale intercept, and a data frame summarizing the path. `coef.ModalNet` returns the named coefficient vector (intercept first), `fitted.ModalNet` the fitted conditional modes, and `logLik.ModalNet` an object of class `"logLik"`.

Examples

```
set.seed(2)
X <- matrix(rnorm(150 * 10), 150, 10)
dat <- generate_modal_data(X, c(0.5, 0.8, -0.6, rep(0, 8)), phi = 0.2,
                           family = "weibull")
fit <- modal_net(dat$y, X, family = "weibull")
print(fit)
s <- summary(fit)
s$coefficients
coef(fit)
head(fitted(fit))
logLik(fit)
```

`modal_net` *Penalized parametric modal regression for positive responses*

Description

Fits the modal regression model $\log M_i = \gamma_0 + x_i^\top \gamma$, where M_i is the conditional mode of a positive response, by maximizing the penalized log-likelihood

$$\ell_p(\gamma, \phi) = \ell(\gamma, \phi)/n - \lambda\, \Psi(\gamma),$$

with Ψ the Lasso, elastic-net, ridge or adaptive-Lasso penalty on the slopes (the intercept is never penalized). The Gamma, Weibull and Lognormal families are available, each written in terms of its mode through the reparameterization of Galarza and Lachos (2026).

Usage

```
modal_net(
  y,
  X,
  family = c("gamma", "weibull", "lognormal"),
  lambda = NULL,
  nlambda = 50,
  lambda.min.ratio = NULL,
  alpha = 1,
  penalty.factor = NULL,
  gamma_ebic = 0.5,
  dfmax = NULL,
  line_search = TRUE,
  standardize = TRUE,
  tol = 1e-06,
  maxit = 200,
  verbose = FALSE
)
```

Arguments

`y`	numeric vector of positive responses.
`X`	numeric matrix (or data frame) of covariates, $n \times p$, without an intercept column.
`family`	one of `"gamma"`, `"weibull"`, `"lognormal"`.
`lambda`	optional user-supplied decreasing sequence of penalty values. If `NULL` (default) a grid of length `nlambda` is built.
`nlambda`	length of the automatic grid.
`lambda.min.ratio`	smallest grid value as a fraction of the largest; default `0.01` when $n > p$ and `0.05` otherwise.
`alpha`	elastic-net mixing parameter passed to **glmnet**: `alpha = 1` (default) is the Lasso, `0 < alpha < 1` the elastic net and `alpha = 0` ridge regression. Ridge produces no exact zeros, so `selected` contains every covariate and the EBIC, which counts active coefficients, is not a selection criterion there; compare ridge fits by prediction instead.
`penalty.factor`	optional vector of length p of relative penalty weights (adaptive Lasso); zeros leave a variable unpenalized. As in **glmnet**, the weights are rescaled to sum to p.
`gamma_ebic`	the EBIC constant $\gamma_{\mathrm{EBIC}} \in [0, 1]$; `0` gives the ordinary BIC.
`dfmax`	the path stops once more than `dfmax` covariates are active; default `min(p, 0.8 n)`. Near saturation ($p \geq n$) the Gamma and Weibull likelihoods are unbounded as $\phi \to 0$, so the dense end of the path carries no information.
`line_search`	logical; use the backtracking safeguard $\alpha^{(k)} \in \{1, 1/2, 1/4, \dots\}$ so that the penalized objective never increases.
`standardize`	passed to **glmnet**; standardize the columns of `X` internally before penalizing.
`tol`	relative tolerance on the penalized objective.
`maxit`	maximum number of proximal Newton iterations per grid point.
`verbose`	logical; report progress along the path through `message`.

Details

The estimator is computed by a *proximal Newton* algorithm, equivalently an iteratively reweighted penalized least-squares (IRPLS) scheme. At each iteration the modal log-likelihood is replaced by its second-order expansion in the linear predictor η_i, which yields a pseudo-response z_i and a working weight w_i in closed form:

- Gamma: $w_i = y_i/(\phi M_i)$, $z_i = \eta_i + 1 - (1+\phi)M_i/y_i$;
- Weibull: $w_i = k^2 (y_i/\lambda_i)^k$, $z_i = \eta_i + \{1 - (\lambda_i/y_i)^k\}/k$;
- Lognormal: $w_i = 1/\phi$, $z_i = \log y_i - \phi$ (the surrogate is exact).

The update of γ is then the weighted penalized least-squares problem $\min_\gamma \frac{1}{2}\sum_i w_i (z_i - x_i^\top \gamma)^2 + \lambda \Psi(\gamma)$, solved exactly by the cyclical coordinate descent of **glmnet** with observation weights, so that variable selection is performed by the optimizer itself (exact zeros, no thresholding). An optional backtracking line search keeps the penalized objective monotone. The dispersion ϕ is

then updated by maximizing the profile log-likelihood (closed form for the Lognormal family, one-dimensional optimization otherwise).

The regularization parameter is chosen by path following: a decreasing grid $\lambda_1 > \cdots > \lambda_L$ is built once from the initial pseudo-response, the algorithm is run to convergence at each grid point with a warm start at the previous solution, and the extended BIC

$$\mathrm{EBIC}(\lambda) = -2\ell + \mathrm{df}\log n + 2\gamma_{\mathrm{EBIC}}\,\mathrm{df}\log p, \qquad \mathrm{df} = |\hat{S}_\lambda| + 2,$$

is evaluated across the converged fits. The initialization uses a lightly penalized Gaussian Lasso of $\log y$ on X, which stays well defined when $p > n$. With an automatic grid, the null model (intercept and dispersion only) is always among the candidates: when the first grid point is not the null model, the null fit is computed directly and placed at the top of the path.

The penalty is $\lambda\Psi(\gamma) = \lambda\sum_j v_j\{\alpha|\gamma_j| + (1-\alpha)\gamma_j^2/2\}$ on the standardized columns (v_j given by `penalty.factor`), and λ is expressed on the scale used by **glmnet**, whose weights are rescaled to sum to n. For the Lasso this makes the Lognormal fit at penalty λ coincide with the Gaussian Lasso of $\log y$ at the *same* λ (slopes identical, intercepts differing by exactly ϕ). For `alpha < 1`, **glmnet** divides its ridge term by the standard deviation of the response it receives; the inner call compensates for this, so the penalty above holds exactly for every `alpha`.

Value

An object of class `"ModalNet"`, a list with components

`gamma`	estimated coefficients (intercept first) at the EBIC-selected λ.
`phi`	estimated dispersion at the selected λ.
`selected`	indices of the selected (nonzero) covariates.
`lambda`	selected penalty.
`loglik`, `ebic`, `df`	values at the selected penalty.
`path`	list with the whole path: `lambda`, `coef` (matrix $(p+1)\times L$), `phi`, `loglik`, `df`, `ebic`, `iter`, `converged`.
`fitted.mode`	fitted conditional modes $\hat{M}_i$.
`mean_const`	the constant $c(\hat{\phi})$ with $E[Y \mid x] = c(\phi)M$; the conditional mean has the same slopes and intercept $\hat{\gamma}_0 + \log c(\hat{\phi})$.
`boundary`	`TRUE` when the dispersion estimate sits at the limit of its range ($10^{-4} \le \phi \le 10^4$), which for the Gamma and Weibull families means a fitted density close to the exponential limit with mode near zero; a warning is issued.

plus `family`, `n`, `p`, `gamma_ebic`, `call`.

References

Galarza, C. E. and Lachos, V. H. (2026). A proximal Newton algorithm for penalized modal regression with high-dimensional positive responses.

Galarza, C. E. and Lachos, V. H. (2026). Parametric modal regression for right-censored positive responses. *arXiv preprint* doi:10.48550/arXiv.2603.07099.

Friedman, J., Hastie, T. and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1), 1–22. doi:10.18637/jss.v033.i01.

Chen, J. and Chen, Z. (2008). Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*, 95, 759–771. doi:10.1093/biomet/asn034.

Examples

```
set.seed(1)
n <- 200; p <- 30
X <- matrix(rnorm(n * p), n, p)
gamma_true <- c(1, 0.6, -0.5, 0.4, rep(0, p - 3))
dat <- generate_modal_data(X, gamma_true, phi = 0.3, family = "gamma")
fit <- modal_net(dat$y, X, family = "gamma")
fit
summary(fit)
plot(fit)
```

`plot.ModalNet` *Plot the EBIC curve and the coefficient path of a ModalNet fit*

Description

Plot the EBIC curve and the coefficient path of a ModalNet fit

Usage

```
## S3 method for class 'ModalNet'
plot(x, which = c(1, 2), ...)
```

Arguments

`x`	a `"ModalNet"` object.
`which`	`1` for the EBIC curve, `2` for the coefficient path, `c(1, 2)` (default) for both side by side.
`...`	further graphical arguments (ignored).

Value

The `"ModalNet"` object, invisibly. Called for its side effect: the EBIC against $\log \lambda$, with the selected value marked, and the coefficient paths on the log-mode scale.

Examples

```
set.seed(4)
X <- matrix(rnorm(150 * 10), 150, 10)
dat <- generate_modal_data(X, c(1, 0.6, -0.4, rep(0, 8)), phi = 0.3,
                           family = "lognormal")
fit <- modal_net(dat$y, X, family = "lognormal")
plot(fit)
plot(fit, which = 2)
```

predict.ModalNet *Predict conditional modes (or means) from a fitted ModalNet model*

Description

Predict conditional modes (or means) from a fitted ModalNet model

Usage

```
## S3 method for class 'ModalNet'
predict(object, newdata, type = c("mode", "link", "mean"), ...)
```

Arguments

object	a "ModalNet" object.
newdata	covariate matrix with the same columns as the training X; if missing, the training matrix is used.
type	"mode" (default) returns $\hat{M}$, "link" the linear predictor $\log \hat{M}$, and "mean" the implied conditional mean $c(\hat{\phi})\hat{M}$.
...	unused.

Value

A numeric vector with one prediction per row of newdata (or of the training matrix), on the scale chosen by type.

Examples

```
set.seed(3)
X <- matrix(rnorm(120 * 8), 120, 8)
dat <- generate_modal_data(X, c(1, 0.7, -0.5, rep(0, 6)), phi = 0.3,
                           family = "gamma")
fit <- modal_net(dat$y[1:100], X[1:100, ], family = "gamma")
new <- X[101:120, ]
cbind(mode = predict(fit, new), mean = predict(fit, new, type = "mean"))[1:5, ]
```

riboflavinV100 *Riboflavin production in Bacillus subtilis (100 most variable genes)*

Description

Riboflavin (vitamin B2) production rate of *Bacillus subtilis* together with the expression levels of the 100 genes with the largest empirical variance, for $n = 71$ fermentation runs. This is the riboflavinV100 subset used by Schelldorfer et al. (2011) and in Galarza and Lachos (2026), derived from the riboflavin data of Bühlmann, Kalisch and Meier (2014) distributed with the **hdi** package.

Usage

```
riboflavinV100
```

Format

A list with two components:

- `y` numeric vector of length 71: riboflavin production rate on the original (positive) scale, obtained as `exp()` of the log-production rate of the **hdi** data.
- `x` numeric matrix 71×100 of log gene expression levels (column names are the gene identifiers).

Details

The 71 runs are repeated measurements on 28 strains; the analyses in the literature, and the one in the package examples, treat them as independent. The model is scale equivariant, so `y` may be rescaled (for instance divided by its mean) for numerical convenience without changing the selected set or the slopes.

Source

Bühlmann, P., Kalisch, M. and Meier, L. (2014). High-dimensional statistics with a view towards applications in biology. *Annual Review of Statistics and Its Application*, 1, 255–278. doi:10.1146/annurevstatistics022513115545. Data from the R package **hdi** (GPL).

References

Schelldorfer, J., Bühlmann, P. and van de Geer, S. (2011). Estimation for high-dimensional linear mixed-effects models using L1-penalization. *Scandinavian Journal of Statistics*, 38, 197–214. doi:10.1111/j.14679469.2011.00740.x.

Examples

```
data(riboflavinV100)
y <- riboflavinV100$y / mean(riboflavinV100$y)
fit <- modal_net(y, riboflavinV100$x, family = "gamma")
colnames(riboflavinV100$x)[fit$selected]
```

Index

∗ **datasets**
riboflavinV100, 8

coef.ModalNet *(*ModalNet-methods*)*, 3

fitted.ModalNet *(*ModalNet-methods*)*, 3

generate_modal_data, 2

logLik.ModalNet *(*ModalNet-methods*)*, 3

message, *5*
modal_net, *2*, *3*, 4
ModalNet-methods, 3

plot.ModalNet, 7
predict.ModalNet, 8
print.ModalNet *(*ModalNet-methods*)*, 3
print.summary.ModalNet
*(*ModalNet-methods*)*, 3

riboflavinV100, 8

summary.ModalNet *(*ModalNet-methods*)*, 3