Package ‘gllvm’

September 18, 2023

Type Package

Title Generalized Linear Latent Variable Models

Version 1.4.3

Date 2023-9-18

Maintainer Jenni Niku <jenni.m.e.niku@jyu.fi>

Description Analysis of multivariate data using generalized linear latent variable models (gllvm). Estimation is performed using either Laplace approximation method or variational approximation method implemented via TMB (Kristensen et al., (2016), <doi:10.18637/jss.v070.i05>). For details see Niku et al. (2019a) <doi:10.1371/journal.pone.0216129> and Niku et al. (2019b) <doi:10.1111/2041-210X.13303>.

License GPL-2

Imports MASS, Matrix, statmod, fishMod, mgcv, alabama, nloptr, methods

Depends R (>= 3.5.0), TMB, mvabund

Encoding UTF-8

LazyData true

LinkingTo TMB, RcppEigen

RoxygenNote 7.2.3

NeedsCompilation yes

URL https://github.com/JenniNiku/gllvm

BugReports https://github.com/JenniNiku/gllvm/issues

Suggests knitr, rmarkdown, testthat, gclus, corrplot, lattice

VignetteBuilder knitr

Author Jenni Niku [aut, cre],
Wesley Brooks [aut],
Riki Herliansyah [aut],
Francis K.C. Hui [aut],
Pekka Korhonen [aut],
Sara Taskinen [aut],
Bert van der Veen [aut],
David I. Warton [aut]

Repository CRAN

Date/Publication 2023-09-18 15:00:02 UTC

R topics documented:

AICc *Corrected Akaike information criterion and number of observations*

Description

Calculates corrected Akaike information criterion for small sample sizes, and extracts number of observations.

Usage

```
## S3 method for class 'gllvm'
AICc(object, ...)

## S3 method for class 'gllvm'
nobs(object, ...)
```

Arguments

object	an object of class ’gllvm’.
...	Not used.

Author(s)

Jenni Niku, Bert van der Veen

anova.gllvm *Analysis Of Deviance for gllvm*

Description

Computes an analysis of deviance table for two generalized linear latent variable model fits.

Usage

```
## S3 method for class 'gllvm'
anova(object, ...)
```

Arguments

object	an object of class ’gllvm’.
...	one or more objects of class ’gllvm’

Details

Computes likelihood-ratio test for two or more gllvm models. Test results makes sense only for nested models. Notice also that this test is not designed for testing models which have degrees of freedom difference larger than 20. For such models the P-value should be treated as very approximate.

Author(s)

Jenni Niku

Examples

```
## Load a dataset from the mvabund package
data(antTraits)
y <- antTraits$abund
X <- antTraits$env
TR <- antTraits$traits
# Fit gllvm model
fit1 <- gllvm(y, X, TR, formula = ~ Bare.ground + Shrub.cover, family = poisson())
fit2 <- gllvm(y, X, TR, formula = ~ Bare.ground + Shrub.cover +
             (Bare.ground + Shrub.cover) : Webers.length, family = poisson())
```

```
# Test if the model with fourth corner interaction terms is significantly
# better using likelihood-ratio test:
anova(fit1, fit2)
```

`coefplot.gllvm` *Plot covariate coefficients and confidence intervals*

Description

Plots covariate coefficients and their confidence intervals.

Usage

```
## S3 method for class 'gllvm'
coefplot(
  object,
  y.label = TRUE,
  which.Xcoef = NULL,
  order = TRUE,
  cex.ylab = 0.5,
  cex.xlab = 1.3,
  mfrow = NULL,
  mar = c(4, 6, 2, 1),
  xlim.list = NULL,
  ...
)
```

Arguments

`object` an object of class 'gllvm'.

`y.label` logical, if `TRUE` (default) colnames of y with respect to coefficients are added to plot.

`which.Xcoef` vector indicating which covariate coefficients will be plotted. Can be vector of covariate names or numbers. Default is `NULL` when all covariate coefficients are plotted.

`order` logical, whether or not coefficients are ordered, defaults to `TRUE`.

`cex.ylab` the magnification to be used for axis annotation relative to the current setting of cex.

`cex.xlab` the magnification to be used for axis annotation.

`mfrow` same as `mfrow` in `par`. If `NULL` (default) it is determined automatically.

`mar` vector of length 4, which defines the margin sizes: `c(bottom, left, top, right)`. Defaults to `c(4,5,2,1)`.

`xlim.list` list of vectors with length of two to define the intervals for an x axis in each covariate plot. Defaults to NULL when the interval is defined by the range of point estimates and confidence intervals

`...` additional graphical arguments.

Author(s)

Jenni Niku <jenni.m.e.niku@jyu.fi>, Francis K.C. Hui, Sara Taskinen, Bert van der Veen

Examples

```
# Extract subset of the microbial data to be used as an example
data(microbialdata)
X <- microbialdata$Xenv
y <- microbialdata$Y[, order(colMeans(microbialdata$Y > 0),
                     decreasing = TRUE)[21:40]]
fit <- gllvm(y, X, formula = ~ pH + Phosp, family = poisson())
coefplot(fit)
## Not run:
## Load a dataset from the mvabund package
data(antTraits)
y <- as.matrix(antTraits$abund)
X <- as.matrix(antTraits$env)
# Fit model with environmental covariates
fit <- gllvm(y, X, formula = ~ Bare.ground + Shrub.cover,
            family = poisson())
coefplot.gllvm(fit)

# Fit model with all environmental covariates
fitx <- gllvm(y, X, family = "negative.binomial")
coefplot(fitx, mfrow = c(3,2))
coefplot(fitx, which.Xcoef = 1:2)

# Fit gllvm model with environmental and trait covariates
TR <- antTraits$traits
fitT <- gllvm(y = y, X = X, TR = TR, family = "negative.binomial")
coefplot(fitT)

# Fit  gllvm model with environmental covariances and reduced rank
fitRR <- gllvm(y = y, X = X, num.RR = 2, family = "negative.binomial")
coefplot(fitRR)

## End(Not run)
```

confint.gllvm *Confidence intervals for model parameters*

Description

Computes confidence intervals for parameters in a fitted gllvm model.

Usage

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

Arguments

object an object of class ’gllvm’.

parm a specification of which parameters are to be given confidence intervals, a vector of names. Examples of options are "beta0", "Xcoef",theta", "phi". If missing, all parameters are considered.

level the confidence level. Scalar between 0 and 1.

... not used.

Author(s)

Jenni Niku <jenni.m.e.niku@jyu.fi>

Examples

```
## Not run:
## Load a dataset from the mvabund package
data(antTraits)
y <- as.matrix(antTraits$abund)
X <- as.matrix(antTraits$env[,1:2])
# Fit gllvm model
fit <- gllvm(y = y, X = X, family = poisson())
# 95 % confidence intervals for coefficients of X variables
confint(fit, level = 0.95, parm = "Xcoef")

## End(Not run)
```

ecoCoefs *Functions to extract ecological quantities of the latent variables from a GLLVM, if species are a quadratic function of the latent variables.*

Description

Extracts species optima and tolerances, potentially with standard errors (derived with the Delta method).

Usage

```
## S3 method for class 'gllvm'
optima(object, sd.errors = TRUE, ...)

## S3 method for class 'gllvm'
tolerances(object, sd.errors = TRUE, ...)
```

Arguments

object an object of class ’gllvm’.

sd.errors logical. If `TRUE`, also returns standard errors.

... Not used.

Details

Currently no separate method for calculating species maxima or gradient length are implemented. Gradient length can be inferred from the standard deviation of the latent variables, which is reported by `summary.gllvm`.

Author(s)

Bert van der Veen

`eSpider` *Hunting spider data*

Description

Extended dataset of counts of hunting spiders in a dune area in the Netherlands, measured at 100 pitfall traps.

Usage

```
data(eSpider)
```

Format

abund A data frame with abundances of 12 hunting spider species measured at 100 sites.

X A matrix of 26 predictor variables at 28 of the 100 sites.

nonNA An vector of indices indicating at which sites the predictor variables were measured.

Details

Counts of hunting spiders in a dune area in the Netherlands, measured with 100 different pitfall traps. This dataset was published with permission from the CANOCO FORTRAN package (version 4 or higher) example datasets.

Species names have been abbreviated, corresponding to: Alopacce = *Alopecosa accentuata*, Alopcune = *Alopecosa cuneata*, Alopfabr = *Alopecosa fabrilis*, Arctlute = *Arctosa lutetiana*, Arctperi = *Arctosa perita*, Auloalbi = *Alonia albimana*, Pardlugu = *Pardosa lugubris*, Pardmont = *Pardosa monticola*, Pardnigr = *Pardosa nigriceps*, Pardpull = *Pardosa pullata*, Trocterr = *Trochosa terricola*, Zoraspin = *Zora spinimana*.

Environmental measurements were taken at 28 of the 100 pitfall traps measuring soil properties(Water content: "conWate", Humus content: "conHumu", Acidity (pH-KCl)), vegetation ("BareSand": percentage bare sand, "FallTwig": cover on the ground by leaves and twigs, "CovMoss": cover by mosses and lichens, "CovHerb": cover by the herb and grass layer (including maximum height, minimum height, "CovCala": cover by *Calamagrostis epigejos*, cover by *Carex arenaria*, "CovFest": cover by *Festuca ovina*, "CovCory": cover by *Corynephorus canescens*, "CovUrti": cover by *Urtica dioica*, "CovMoeh": cover by *Moehringia trinervia*), "CovShru": cover by the shrub layer (minimum and maximum height, and "CovLigu": cover by *Ligustrum vulgare*), "CovTree": cover by the tree layer (including maximum height, cover by *Populus tremula* or *Crataegus* *monogyna*)), and light properties ("LuxGrey": lux at equal grey sky, "LuxSun": lux at cloudless sky, "LuxRef": lux by reflection of the soil surface)

The original publication of Canonical Correspondence Analysis used standardized versions of the `log` and `log1p` transformed predictors "ConWate", "BareSand", "FallTwig", "CovMoss", "CovHerb", "LuxRef".

References

ter Braak, C.J.F. and Smilauer, P. (1998). CANOCO reference manual and user's guide to CANOCO for Windows: software for canonical community ordination (version 4). Microcomputer Power, New York, New York, USA. ter Braak, C.J.F. (1986). Canonical correspondence analysis: a new eigenvector technique for multivariate direct gradient analysis. Ecology, 67(5), 1167-1179. Van der Aart, P. J. M. and Smeenk-Enserink, N. (1975). Correlations between distributions of hunting spiders (*Lycosidae*, *Ctenidae*) and environmental characteristics in a dune area. Netherlands Journal of Zoology, 25(1), 1-45.

Examples

```
data(eSpider)
Y <- eSpider$abund[eSpider$nonNA, ]
X <- eSpider$X[eSpider$nonNA, ]
model <- gllvm(y = Y, X = X,
    lv.formula = ~log(ConWate) + log1p(BareSand) + log1p(FallTwig) +
    log1p(CovMoss) + log1p(CovHerb) + log(LuxRef),
    num.RR = 2,
    family = "poisson")
```

`getLV.gllvm` *Extract latent variables*

Description

Extract latent variables from gllvm object.

Usage

```
## S3 method for class 'gllvm'
getLV(object, type = NULL, ...)
```

Arguments

`object`	an object of class 'gllvm'.
`type`	type of latent variable scores to retrieve from a gllvm object. For models with unconstrained latent variables, defaults to "residual". For models with constrained latent variables, defaults to conditional. Alternatively, "marginal" returns linear combination scores without residual error.
`...`	not used

Details

Function retrieves the site scores for a GLLVM. Each type corresponds to a separate term of the model. For a GLLVM with unconstrained latent variables the default is "residual". "Residual" scores represent the error term in concurrent ordination, and are not available for constrained ordination.

For GLLVMs with informed latent variables, "conditional" returns the complete site scores, due to both fixed- and latent effects, where the latent effect is always scaled by the diagonal of the species loadings so that it can be small relative to the fixed-effects. "Conditional" here means conditional on the random-effect i.e. the residual.

Type "marginal" returns linear combination scores, i.e. the site scores only due to fixed-effects. These are available for constrained and concurrent ordination.

If both unconstrained and constrained latent variables are included in the model, type "marginal" returns linear combination scores for constrained latent variables but "residual" scores for unconstrained latent variables.

getPredictErr.gllvm *Extract prediction errors for latent variables from gllvm object*

Description

Calculates the prediction errors for latent variables and random effects for gllvm model.

Usage

```
## S3 method for class 'gllvm'
getPredictErr(object, CMSEP = TRUE, cov = FALSE, ...)
```

Arguments

`object`	an object of class 'gllvm'.
`CMSEP`	logical, if `TRUE` conditional mean squared errors for predictions are calculated. If `FALSE`, prediction errors are based on covariances of the variational distributions for `method ="VA"` and `method ="EVA"`.
`cov`	if `TRUE`, return as covariances/variances of predictions. Otherwise `FALSE` (default) return as standard errors of predictions.
`...`	not used

Details

Calculates conditional mean squared errors for predictions. If variational approximation is used, prediction errors can be based on covariances of the variational distributions, and therefore they do not take into account the uncertainty in the estimation of (fixed) parameters.

Value

Function returns following components:

`lvs`	prediction errors for latent variables
`row.effects`	prediction errors for random row effects if included

Author(s)

Francis K.C. Hui, Jenni Niku, David I. Warton

Examples

```
## Not run:
# Load a dataset from the mvabund package
data(antTraits)
y <- as.matrix(antTraits$abund)
# Fit gllvm model
fit <- gllvm(y = y, family = poisson())
# prediction errors for latent variables:
getPredictErr(fit)

## End(Not run)
```

`getResidualCor.gllvm` *Extract residual correlations from gllvm object*

Description

Calculates the residual correlation matrix for gllvm model.

Usage

```
## S3 method for class 'gllvm'
getResidualCor(object, adjust = 1, site.index = NULL, ...)
```

Arguments

`object` an object of class 'gllvm'.

`adjust` The type of adjustment used for negative binomial and binomial distribution when computing residual correlation matrix. Options are 0 (no adjustment), 1 (the default adjustment) and 2 (alternative adjustment for NB distribution). See details.

`site.index` A site index used used in the calculation of a GLLVM with quadratic response model, for which the residual correlations are calculated.

`...` not used

Details

Residual correlation matrix is calculated based on the residual covariance matrix, see details from getResidualCov.gllvm.

Author(s)

Francis K.C. Hui, Jenni Niku, David I. Warton

Examples

```
#'# Extract subset of the microbial data to be used as an example
data(microbialdata)
y <- microbialdata$Y[, order(colMeans(microbialdata$Y > 0),
                     decreasing = TRUE)[21:40]]
fit <- gllvm(y, family = poisson())
fit$logL
cr <- getResidualCor(fit)
cr[1:5,1:5]
## Not run:
# Load a dataset from the mvabund package
data(antTraits)
y <- as.matrix(antTraits$abund)
# Fit gllvm model
fit <- gllvm(y = y, family = poisson())
# residual correlations:
cr <- getResidualCor(fit)
# Plot residual correlations:
install.packages("corrplot", "gclus")
library(corrplot)
library(gclus)
corrplot(cr[order.single(cr), order.single(cr)], diag = F,
  type = "lower", method = "square", tl.cex = 0.8, tl.srt = 45, tl.col = "red")

## End(Not run)
```

---

getResidualCov.gllvm *Extract residual covariance matrix from gllvm object*

Description

Calculates the residual covariance matrix for gllvm model.

Usage

```
## S3 method for class 'gllvm'
getResidualCov(object, adjust = 1, site.index = NULL, ...)
```

Arguments

`object` an object of class 'gllvm'.

`adjust` The type of adjustment used for negative binomial, binomial and normal distribution when computing residual correlation matrix. Options are 0 (no adjustment), 1 (the default adjustment) and 2 (alternative adjustment for NB distribution), see details.

`site.index` A site index, vector of length one or 1, that is used in the calculation of a GLLVM with quadratic response model.

`...` not used.

Details

Residual covariance matrix, storing information on species co-occurrence that is not explained by the environmental variables (if included), is calculated using the matrix of latent variables loadings, that is, $\Theta\Theta'$, and the dispersion parameter related to the distribution of choice, is applicable (e.g. in the case of negative-binomial distributed responses).

When the responses are modelled using the negative binomial distribution, the residual variances for each species must be adjusted for overdispersion. The two possible adjustment terms are $log(\phi_j+1)$ (`adjust = 1`) and $\psi^{(1)}(1/\phi_j)$ (`adjust = 2`), where $\psi^{(1)}$ is the trigamma function.

The negative binomial model can be written using different parameterizations. The residual covariance with `adjust = 1` can be obtained using the lognormal-Poisson parametrization, that is,

$$Y_{ij} \sim Poisson(\mu_{ij}\lambda_j),$$

where $\lambda_j \sim lognormal(-\sigma^2/2, \sigma^2)$ and $\sigma^2 = log(\phi_j + 1)$ and $log(\mu_{ij}) = \eta_{ij}$. Now $E[Y_{ij}] = \mu_{ij}$ and variance $V(\mu_{ij}) = \mu_{ij} + \mu_{ij}^2(exp(\sigma^2) - 1) = \mu_{ij} + \mu_{ij}^2\phi_j$, which are the same as for the NB distribution. Therefore, on linear predictor scale, we have the variance

$$V(log(\mu_{ij}\lambda_j)) = V(log\mu_{ij}) + V(log\lambda_j) = V(u_i'\theta_j) + \sigma^2 = \theta_j'\theta_j + log(\phi_j + 1).$$

which leads to the residual covariance matrix $\Theta\Theta' + \Psi$, where Ψ is the diagonal matrix with $log(\phi_j + 1)$ as diagonal elements (`adjust = 1`).

Or, for a GLLVM where species are a quadratic function of the latent variables, we instead have

$$V(log(\mu_{ij}\lambda_j)) = V(log\mu_{ij}) + V(log\lambda_j) = V(u_i'\theta_j - u_i'D_ju_i) + \sigma^2$$
$$= \theta_j'\theta_j + 2diag(D_j)'diag(D_j)log(\phi_j + 1).$$

which leads to the residual covariance matrix $\Theta\Theta' + 2\Gamma_j\Gamma_j' + diag(\Phi)$, where Γ_j holds the quadratic coefficients. Since the quadratic coefficients are constrained to be positive, the residual covariance in the latter case is, given the same coefficients on the linear term, equal or more positive than in the linear case.

The residual covariance matrix with `adjust = 2` can be obtained by using Poisson-Gamma parametrization

$$Y_{ij} \sim Poisson(\mu_{ij}\lambda_j),$$

where $\lambda_j \sim Gamma(1/\phi_j, 1/\phi_j)$ and μ_{ij} is as above. The mean and the variance are of similar form as above and we have that

$$V(log(\mu_{ij}\lambda_j)) = V(log\mu_{ij}) + V(log\lambda_j) = \theta_j'\theta_j + \psi^{(1)}(1/\phi_j),$$

where $\psi^{(1)}$ is the trigamma function.

In the case of binomial distribution, the adjustment terms (`adjust = 1`) are 1 for probit link and $\pi^2/3$ for logit link. These are obtained by treating binomial model as latent variable model. Assume

$$Y^*_{ij} = \eta_{ij} + e_{ij},$$

where $e_{ij} \sim N(0, 1)$ for probit model, and $e_{ij} \sim logistic(0, 1)$ for logit model. Then binary response is defined as $Y_{ij} = 1$, if $Y^*_{ij} > 0$ and 0 otherwise. Now we have that $\mu_{ij} = P(Y_{ij} = 1) = P(Y^*_{ij} > 0) = P(\eta_{ij} > -e_{ij}) = P(e_{ij} <= \eta_{ij})$ which leads to probit and logit models. On linear predictor scale we then have that

$$V(\eta_{ij} + e_{ij}) = V(\eta_{ij}) + V(e_{ij}).$$

For the probit model, the residual covariance matrix is then $\Theta\Theta' + I_m$, and for the logit model $\Theta\Theta' + \pi^2/3I_m$. Similarly as above, for a GLLVM where species are a quadratic function of the latent variables, the term $2\Gamma_j\Gamma'_j$ is added to the residual covariance matrix.

For normal distribution, we can write

$$Y_{ij} = \eta_{ij} + e_{ij},$$

where $e_{ij} \sim N(0, \phi_j^2)$ and thus we have that

$$V(\eta_{ij} + e_{ij}) = V(\eta_{ij}) + V(e_{ij}).$$

For the gaussian model, the residual covariance matrix is then $\Theta\Theta' + diag(\Phi^2)$.

Value

Function returns following components:

`cov`	residual covariance matrix
`trace`	trace of the residual covariance matrix, the total variance explained
`var.q`	trace of the residual covariance matrix per latent variable, variance explained per latent variable
`var.q2`	trace of the squared term of the residual covariance matrix per latent variable, for quadratic responses. Variance explained per latent variable by the quadratic term

Author(s)

Francis K.C. Hui, Jenni Niku, David I. Warton, Bert van der Veen

Examples

```
## Not run:
# Load a dataset from the mvabund package
data(antTraits)
y <- as.matrix(antTraits$abund)
# Fit gllvm model
fit <- gllvm(y = y, family = poisson())
```

```
# residual covariance:
rescov <- getResidualCov(fit)
rescov$cov
# Trace of the covariance matrix
rescov$trace
# Variance explained per latent variable
rescov$var.q

## End(Not run)
```

gllvm *Generalized Linear Latent Variable Models*

Description

Fits generalized linear latent variable model for multivariate data. The model can be fitted using Laplace approximation method or variational approximation method.

Usage

```
gllvm(
  y = NULL,
  X = NULL,
  TR = NULL,
  data = NULL,
  formula = NULL,
  family,
  num.lv = NULL,
  num.lv.c = 0,
  num.RR = 0,
  lv.formula = NULL,
  lvCor = NULL,
  studyDesign = NULL,
  dist = matrix(0),
  corWithin = FALSE,
  quadratic = FALSE,
  row.eff = FALSE,
  sd.errors = TRUE,
  offset = NULL,
  method = "VA",
  randomB = FALSE,
  randomX = NULL,
  dependent.row = FALSE,
  beta0com = FALSE,
  zeta.struc = "species",
  plot = FALSE,
  link = "probit",
  Ntrials = 1,
  Power = 1.1,
  seed = NULL,
  scale.X = TRUE,
  return.terms = TRUE,
  gradient.check = FALSE,
  disp.formula = NULL,
 control = list(reltol = 1e-10, reltol.c = 1e-08, TMB = TRUE, optimizer = ifelse((num.RR
  + num.lv.c) == 0 | randomB != FALSE, "optim", "alabama"), max.iter = 6000, maxit =
    6000, trace = FALSE, optim.method = NULL),
 control.va = list(Lambda.struc = "unstructured", Ab.struct = "unstructured", Ar.struc =
   "diagonal", diag.iter = 1, Ab.diag.iter = 0, Lambda.start = c(0.3, 0.3, 0.3), NN = 3),
 control.start = list(starting.val = "res", n.init = 1, n.init.max = 10, jitter.var = 0,
   start.fit = NULL, start.lvs = NULL, randomX.start = "zero", quad.start = 0.01,
    start.struc = "LV", scalmax = 10, MaternKappa = 1.5, rangeP = NULL),
  setMap = NULL,
  ...
)
```

Arguments

`y` (n x m) matrix of responses.

`X` matrix or data.frame of environmental covariates.

`TR` matrix or data.frame of trait covariates.

`data` data in long format, that is, matrix of responses, environmental and trait covariates and row index named as "id". When used, model needs to be defined using formula. This is alternative data input for y, X and TR.

`formula` an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted (for fixed-effects predictors).

`family` distribution function for responses. Options are `"negative.binomial"` (with log link), `poisson(link = "log")`, `binomial(link = "probit")` (and also with `link = "logit"` when `method = "LA"` or `method = "EVA"`), zero-inflated poisson (`"ZIP"`), zero-inflated negative-binomial (`"ZINB"`), `gaussian(link = "identity")`, Tweedie (`"tweedie"`) (with log link, for `"LA"` and `"EVA"`-method), `"gamma"` (with log link), `"exponential"` (with log link), beta (`"beta"`) (with logit and probit link, for `"LA"` and `"EVA"`-method) and `"ordinal"` (only with `"VA"`-method).

`num.lv` number of latent variables, d, in gllvm model. Non-negative integer, less than number of response variables (m). Defaults to 2, if `num.lv.c=0` and `num.RR=0`, otherwise 0.

`num.lv.c` number of latent variables, d, in gllvm model to constrain, with residual term. Non-negative integer, less than number of response (m) and equal to, or less than, the number of predictor variables (k). Defaults to 0. Requires specification of "lv.formula" in combination with "X" or "datayx". Can be used in combination with num.lv and fixed-effects, but not with traits.

`num.RR` number of latent variables, d, in gllvm model to constrain, without residual term (reduced rank regression). Cannot yet be combined with traits.

`lv.formula` an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted (for latent variables).

lvCor (Under development, not to be used at the moment!) correlation structure for latent variables, defaults to NULL Correlation structure for latent variables can be defined via formula, eg. ~struc(1|groups), where option to 'struc' are corAR1 (AR(1) covariance), corExp (exponentially decaying, see argument 'dist') and corCS (compound symmetry). The grouping variable needs to be included either in 'X' or 'studyDesign'. Works at the moment only with unconstrained ordination without quadratic term.

studyDesign variables related to eg. sampling/study design, used for defining correlation structure of the latent variables and row effects.

dist matrix of coordinates or time points used for row parameters correlation structure corExp.

corWithin logical. If TRUE, correlation is set between row effects of the observation units within group. Correlation and groups can be defined using row.eff. Defaults to FALSE, when correlation is set for row parameters between groups.

quadratic either FALSE(default), TRUE, or LV. If FALSE models species responses as a linear function of the latent variables. If TRUE models species responses as a quadratic function of the latent variables. If LV assumes species all have the same quadratic coefficient per latent variable.

row.eff FALSE, fixed, "random" or formula to define the structure for the row parameters. Indicating whether row effects are included in the model as a fixed or as a random effects. Defaults to FALSE when row effects are not included. Structured random row effects can be defined via formula, eg. ~(1|groups), when unique row effects are set for each group, not for all rows, the grouping variable needs to be included in X. Correlation structure between random group effects/intercepts can also be set using ~struc(1|groups), where option to 'struc' are corAR1 (AR(1) covariance), corExp (exponentially decaying, see argument 'dist') and corCS (compound symmetry). Correlation structure can be set between or within groups, see argument 'corWithin'.

sd.errors logical. If TRUE (default) standard errors for parameter estimates are calculated.

offset vector or matrix of offset terms.

method model can be fitted using Laplace approximation method (method = "LA") or variational approximation method (method = "VA"), or with extended variational approximation method (method = "EVA") when VA is not applicable. If particular model has not been implemented using the selected method, model is fitted using the alternative method as a default. Defaults to "VA".

randomB either FALSE(default), "LV", "P", or "single". Fits concurrent or constrained ordination (i.e. models with num.lv.c or num.RR) with random slopes for the predictors. "LV" assumes LV-specific variance parameters, "P" predictor specific, and "single" the same across LVs and predictors.

randomX formula for species specific random effects of environmental variables in fourth corner model. Defaults to NULL, when random slopes are not included.

dependent.row logical. Whether or not random row effects are correlated (dependent) with the latent variables. Defaults to FALSE when correlation terms are not included.

beta0com logical. If FALSE column-specific intercepts are assumed. If TRUE, a common intercept is used which is allowed only for fourth corner models.

`zeta.struc` structure for cut-offs in the ordinal model. Either "common", for the same cut-offs for all species, or "species" for species-specific cut-offs. For the latter, classes are arbitrary per species, each category per species needs to have at least one observations. Defaults to "species".

`plot` logical. If `TRUE` ordination plots will be printed in each iteration step when `TMB = FALSE`. Defaults to `FALSE`.

`link` link function for binomial family if `method = "LA"` and beta family. Options are "logit" and "probit.

`Ntrials` number of trials for binomial family.

`Power` fixed power parameter in Tweedie model. Scalar from interval (1,2). Defaults to 1.1. If set to NULL it is estimated (note: experimental).

`seed` a single seed value, defaults to `NULL`.

`scale.X` logical. If `TRUE`, covariates are scaled when fourth corner model is fitted.

`return.terms` logical. If `TRUE` 'terms' object is returned.

`gradient.check` logical. If `TRUE` gradients are checked for large values (>0.01) even if the optimization algorithm did converge.

`disp.formula` formula, or alternatively a vector of indices, for the grouping of dispersion parameters (e.g. in a negative-binomial distribution). Defaults to NULL so that all species have their own dispersion parameter. Is only allowed to include categorical variables. If a formula, data should be included as named rows in y.

`control` A list with the following arguments controlling the optimization:

- *reltol*: convergence criteria for log-likelihood, defaults to 1e-8.
- *reltol.c*: convergence criteria for equality constraints in ordination with predictors, defaults to 1e-8.
- *TMB*: logical, if `TRUE` model will be fitted using Template Model Builder (TMB). TMB is always used if `method = "LA"`. Defaults to `TRUE`.
- *optimizer*: if `TMB=TRUE`, log-likelihood can be optimized using `"optim"` (default) or `"nlminb"`. For ordination with predictors (num.RR>0 or num.lv.c>0) this can additionally be one of `alabama`(default), `nloptr(agl)` or `nloptr(sqp)`.
- *max.iter*: maximum number of iterations when `TMB = FALSE` or for `optimizer = "nlminb"` when `TMB = TRUE`, defaults to 4000.
- *maxit*: maximum number of iterations for optimizer, defaults to 4000.
- *trace*: logical, if `TRUE` in each iteration step information on current step will be printed. Defaults to `FALSE`. Only with `TMB = FALSE`.
- *optim.method*: optimization method to be used if optimizer is `"optim"`,`"alabama"`, or `"nloptr"`, but the latter two are only available in combination with at least two latent variables (i.e., num.RR+num.lv.c>1). Defaults to `"BFGS"`, but to `"L-BFGS-B"` for Tweedie family due the limited-memory use. For optimizer='alabama' this can be any `"optim"` method, or `"nlminb"`. If optimizer = 'nloptr(agl)' this can be one of: "NLOPT_LD_CCSAQ", "NLOPT_LD_SLSQP", "NLOPT_LD_TNEWTON_PRECOND" (default), "NLOPT_LD_TNEWTON", "NLOPT_LD_MMA".

`control.va` A list with the following arguments controlling the variational approximation method:

- *Lambda.struc*: covariance structure of VA distributions for latent variables when `method = "VA"`, "unstructured" or "diagonal".
- *Ab.struct*: covariance structure of VA distributions for random slopes when `method = "VA"`, "unstructured" or "diagonal".
- *Ar.struc*: covariance structure of VA distributions for random row effects when `method = "VA"`, "unstructured" or "diagonal".
- *diag.iter*: non-negative integer which can sometimes be used to speed up the updating of variational (covariance) parameters in VA method. Can sometimes improve the accuracy. If `TMB = TRUE` either 0 or 1. Defaults to 1.
- *Ab.diag.iter*: As above, but for variational covariance of random slopes.
- *Lambda.start*: starting values for variances in VA distributions for latent variables, random row effects and random slopes in variational approximation method. Defaults to 0.3.
- *NN*: Number of nearest neighbors for NN variational covariance. Defaults to ...

`control.start` A list with the following arguments controlling the starting values:

- *starting.val*: starting values can be generated by fitting model without latent variables, and applying factorial analysis to residuals to get starting values for latent variables and their coefficients (`starting.val = "res"`). Another options are to use zeros as a starting values (`starting.val = "zero"`) or initialize starting values for latent variables with (n x num.lv) matrix. Defaults to `"res"`, which is recommended.
- *n.init*: number of initial runs. Uses multiple runs and picks up the one giving highest log-likelihood value. Defaults to 1.
- *n.init.max*: maximum number of refits try try for n.init without improvement, defaults to 10.
- *start.fit*: object of class 'gllvm' which can be given as starting parameters for count data (poisson, NB, or ZIP).
- *start.lvs*: initialize starting values for latent variables with (n x num.lv) matrix. Defaults to `NULL`.
- *jitter.var*: jitter variance for starting values of latent variables. Defaults to 0, meaning no jittering.
- *randomX.start*: starting value method for the random slopes. Options are `"zero"` and `"res"`. Defaults to `"zero"`.
- *start.struc*: starting value method for the quadratic term. Options are `"LV"` (default) and `"all"`.
- *quad.start*: starting values for quadratic coefficients. Defaults to 0.01.
- *MaternKappa*: Starting value for smoothness parameter kappa of Matern covariance function. Defaults to 3/2.
- *scalmax*: Sets starting value for the scale parameter for the coordinates. Defaults to 10, when the starting value for scale parameter scales the distances of coordinates between 0-10.
- *rangeP*: Sets starting value for the range parameter for the correlation structure.

`setMap` UNDER DEVELOPMENT, DO NOT USE! list of a set of parameters to be fixed

... Not used.

Details

Fits generalized linear latent variable models as in Hui et al. (2015 and 2017) and Niku et al. (2017). Method can be used with two types of latent variable models depending on covariates. If only site related environmental covariates are used, the expectation of response Y_{ij} is determined by

$$g(\mu_{ij}) = \eta_{ij} = \alpha_i + \beta_{0j} + x_i'\beta_j + u_i'\theta_j,$$

where $g(.)$ is a known link function, u_i are d-variate latent variables (d«m), α_i is an optional row effect at site i, and it can be fixed or random effect (also other structures are possible, see below), β_{0j} is an intercept term for species j, β_j and θ_j are column specific coefficients related to covariates and the latent variables, respectively.

Quadratic model: Alternatively, a more complex version of the model can be fitted with `quadratic = TRUE`, where species are modeled as a quadratic function of the latent variables:

$$g(\mu_{ij}) = \eta_{ij} = \alpha_i + \beta_{0j} + x_i'\beta_j + u_i'\theta_j - u_i'D_ju_i$$

. Here, D_j is a diagonal matrix of positive only quadratic coefficients, so that the model generates concave shapes only. This implementation follows the ecological theoretical model where species are generally recognized to exhibit non-linear response curves. For a model with quadratic responses, quadratic coefficients are assumed to be the same for all species:

$$D_j = D$$

. This model requires less information per species and can be expected to be more applicable to most datasets. The quadratic coefficients D can be used to calculate the length of ecological gradients. For quadratic responses, it can be useful to provide the latent variables estimated with a GLLVM with linear responses, or estimated with (Detrended) Correspondence Analysis. The latent variables can then be passed to the `start.lvs` argument inside the `control.start` list, which in many cases gives good results.

Ordination with predictors: For GLLVMs with both linear and quadratic response model, a series of predictors x_{lv} can be included to explain the latent variables:

$$g(\mu_{ij}) = \alpha_i + \beta_{0j} + x_i'\beta_j + (B'x_{lv,i} + \epsilon_i)'\gamma_j - (B'x_{lv,i} + \epsilon_i)'D_j(B'x_{lv,i} + \epsilon_i),$$

where $z_i = B'x_{lv,i} + \epsilon_i$ are latent variables informed by the predictors, but not constrained compared to unconstrained ordination as in methods such as CCA or RDA. Omitting the predictors results in an unconstrained ordination, and omitting ϵ_i in the usual constrained ordination, which can also be fitted.

Fourth corner model: An alternative model is the fourth corner model (Brown et al., 2014, Warton et al., 2015) which will be fitted if also trait covariates are included. The expectation of response Y_{ij} is

$$g(\mu_{ij}) = \alpha_i + \beta_{0j} + x_i'(\beta_x + b_j) + TR_j'\beta_t + vec(B) * kronecker(TR_j, X_i) + u_i'\theta_j - u_i'D_ju_i$$

where g(.), u_i, β_{0j} and θ_j are defined as above. Vectors β_x and β_t are the main effects or coefficients related to environmental and trait covariates, respectively, matrix B includes interaction terms. Vectors b_j are optional species-specific random slopes for environmental covariates. The interaction/fourth corner terms are optional as well as are the main effects of trait covariates.

Structured row effects: In addition to the site-specific random effects, α_i, it is also possible to set arbitrary structure/design for the row effects. That is, assume that observations / rows $i = 1, ..., n$ in the data matrix are from groups $t = 1, ..., T$, so that each row i belongs to one of the groups, denote $G(i) \in \{1, ..., T\}$. Each group t has a number of observations n_t, so that $\sum_{t=1}^{T} n_t = n$. Now we can set random intercept for each group t, (see argument 'row.eff'):

$$g(\mu_{ij}) = \eta_{ij} = \alpha_{G(i)} + \beta_{0j} + x_i'\beta_j + u_i'\theta_j,$$

There is also a possibility to set correlation structure for the random intercepts between groups, so that $(\alpha_1, ..., \alpha_T)^\top \sim N(0, \Sigma_r)$. That might be the case, for example, when the groups are spatially or temporally dependent. Another option is to set row specific random intercepts α_i, but to set the correlation structure for the observations within groups, (see argument 'corWithin'). That is, we can set $corr(\alpha_i, \alpha_{i'}) = C(i, i') \neq 0$ according to some correlation function C, when $G(i) = G(i')$. This model is restricted to the case, where each group has equal number of observations (rows), that is $n_t = n_{t'}$ for all $t, t' \in \{1, ..., T\}$.

The correlation structures available in the package are

- corAR1 autoregressive process of order 1.
- corExp exponentially decaying, see argument 'dist'.
- corCS compound symmetry.

Starting values: The method is sensitive for the choices of initial values of the latent variables. Therefore it is recommendable to use multiple runs and pick up the one giving the highest log-likelihood value (see argument 'n.init'). However, sometimes this is computationally too demanding, and default option starting.val = "res" is recommended. For more details on different starting value methods, see Niku et al., (2018).

Models are implemented using TMB (Kristensen et al., 2015) applied to variational approximation (Hui et al., 2017), extended variational approximation (Korhonen et al., 2021) and Laplace approximation (Niku et al., 2017).

With ordinal family response classes must start from 0 or 1.

Distributions:

Mean and variance for distributions are defined as follows.

- For count data family = poisson(): Expectation $E[Y_{ij}] = \mu_{ij}$, variance $V(\mu_{ij}) = \mu_{ij}$, or
- family = "negative.binomial": Expectation $E[Y_{ij}] = \mu_{ij}$, variance $V(\mu_{ij}) = \mu_{ij} + \mu_{ij}^2\phi_j$, or
- family = "ZIP": Expectation $E[Y_{ij}] = (1-p)\mu_{ij}$, variance $V(\mu_{ij}) = \mu_{ij}(1-p_j)(1+\mu_{ij}p)$.
- family = "ZINB": Expectation $E[Y_{ij}] = (1-p)\mu_{ij}$, variance $V(\mu_{ij}) = \mu_{ij}(1-p_j)(1+\mu_{ij}(\phi_j + p_j))$.
- For binary data family = binomial(): Expectation $E[Y_{ij}] = \mu_{ij}$, variance $V(\mu_{ij}) = \mu_{ij}(1-\mu_{ij})$.
- For percent cover data $0 < Y_{ij} < 1$ family = "beta": Expectation $E[Y_{ij}] = \mu_{ij}$, variance $V(\mu_{ij}) = \mu_{ij}(1-\mu_{ij})//(1+\phi_j)$.

- For positive continuous data `family = "gamma"`:Expectation $E[Y_{ij}] = \mu_{ij}$, variance $V(\mu_{ij}) = \mu_{ij}^2/\phi_j$, where ϕ_j is species specific shape parameter.
- For non-negative continuous data `family = "exponential"`:Expectation $E[Y_{ij}] = \mu_{ij}$, variance $V(\mu_{ij}) = \mu_{ij}^2$.
- For non-negative continuous or biomass data `family = "tweedie"` Expectation $E[Y_{ij}] = \mu_{ij}$, variance $V(\mu_{ij}) = \phi_j * \mu_{ij}^\nu$, where ν is a power parameter of Tweedie distribution. See details Dunn and Smyth (2005).
- For ordinal data `family = "ordinal"`: Cumulative probit model, see Hui et.al. (2016).
- For normal distributed data `family = gaussian()`: Expectation $E[Y_{ij}] = \mu_{ij}$, variance $V(y_{ij}) = \phi_j^2$.

Value

An object of class "gllvm" includes the following components:

`call`	function call
`y`	(n x m) matrix of responses.
`X`	matrix or data.frame of environmental covariates.
`lv.X`	matrix or data.frame of environmental covariates for latent variables.
`TR`	Trait matrix
`formula`	Formula for predictors
`lv.formula`	Formula of latent variables in constrained and concurrent ordination
`randomX`	Formula for species specific random effects in fourth corner model
`randomB`	Boolean flag for random slopes in constrained and concurrent ordination
`num.lv`	Number of unconstrained latent variables
`num.lv.c`	Number of latent variables in concurrent ordination
`num.RR`	Number of latent variables in constrained ordination
`Ntrials`	Number of trials in a binomial model
`method`	Method used for integration
`family`	Response distribution
`row.eff`	Type of row effect used
`n.init`	Number of model runs for best fit
`disp.group`	Groups for dispersion parameters
`sd`	List of standard errors
`lvs`	Latent variables
`params`	List of parameters • theta latent variables' loadings relative to the diagonal entries of loading matrix • sigma.lv diagonal entries of latent variables' loading matrix • LvXcoef Predictor coefficients (or predictions for random slopes) related to latent variables, i.e. canonical coefficients

- beta0 column specific intercepts
- Xcoef coefficients related to environmental covariates X
- B coefficients in fourth corner model
- row.params row-specific intercepts
- phi dispersion parameters ϕ for negative binomial or Tweedie family, probability of zero inflation for ZIP family, standard deviation for gaussian family or shape parameter for gamma/beta family
- inv.phi dispersion parameters $1/\phi$ for negative binomial

`Power` power parameter ν for Tweedie family

`sd` list of standard errors of parameters

`prediction.errors`
list of prediction covariances for latent variables and variances for random row effects when method `"LA"` is used

`A, Ar, Ab_lv` covariance matrices for variational densities of latent variables, random row effects, and random slopes respectively

`seed` Seed used for calculating starting values

`TMBfn` TMB objective and derivative functions

`logL` log likelihood

`convergence` convergence code of optimizer

`quadratic` flag for quadratic model

`Hess` List holding matrices of second derivatives

`beta0com` Flag for common intercept in fourth corner models

`rstruc` Integer that indicates which type of row structure is included

`cstruc` Correlation structure for row effects

`dist` Matrix of coordinates or time points used for row effects

`terms` Terms object for main predictors

`start` starting values for model

`optim.method` Optimization method when using 'optim', 'alabama', or 'nloptr'

Note

If function gives warning: 'In f(x, order = 0) : value out of range in 'lgamma'', optimizer have visited an area where gradients become too big. It is automatically fixed by trying another step in the optimization process, and can be ignored if errors do not occur.

Author(s)

Jenni Niku <jenni.m.e.niku@jyu.fi>, Wesley Brooks, Riki Herliansyah, Francis K.C. Hui, Pekka Korhonen, Sara Taskinen, Bert van der Veen, David I. Warton

References

Brown, A. M., Warton, D. I., Andrew, N. R., Binns, M., Cassis, G., and Gibb, H. (2014). The fourth-corner solution - using predictive models to understand how species traits interact with the environment. Methods in Ecology and Evolution, 5:344-352.

Dunn, P. K. and Smyth, G. K. (2005). Series evaluation of tweedie exponential dispersion model densities. Statistics and Computing, 15:267-280.

Hui, F. K. C., Taskinen, S., Pledger, S., Foster, S. D., and Warton, D. I. (2015). Model-based approaches to unconstrained ordination. Methods in Ecology and Evolution, 6:399-411.

Hui, F. K. C., Warton, D., Ormerod, J., Haapaniemi, V., and Taskinen, S. (2017). Variational approximations for generalized linear latent variable models. Journal of Computational and Graphical Statistics. Journal of Computational and Graphical Statistics, 26:35-43.

Kasper Kristensen, Anders Nielsen, Casper W. Berg, Hans Skaug, Bradley M. Bell (2016). TMB: Automatic Differentiation and Laplace Approximation. Journal of Statistical Software, 70(5), 1-21.

Korhonen, P., Hui, F. K. C., Niku, J., and Taskinen, S. (2021). Fast, universal estimation of latent variable models using extended variational approximations. Stat Comput 33, 26 (2023).

Niku, J., Warton, D. I., Hui, F. K. C., and Taskinen, S. (2017). Generalized linear latent variable models for multivariate count and biomass data in ecology. Journal of Agricultural, Biological, and Environmental Statistics, 22:498-522.

Niku, J., Brooks, W., Herliansyah, R., Hui, F. K. C., Taskinen, S., and Warton, D. I. (2018). Efficient estimation of generalized linear latent variable models. PLoS One, 14(5):1-20.

Warton, D. I., Guillaume Blanchet, F., O'Hara, R. B., Ovaskainen, O., Taskinen, S., Walker, S. C. and Hui, F. K. C. (2015). So many variables: Joint modeling in community ecology. Trends in Ecology & Evolution, 30:766-779.

See Also

coefplot.gllvm, confint.gllvm, ordiplot.gllvm, plot.gllvm, summary.gllvm.

Examples

```
# Extract subset of the microbial data to be used as an example
data(microbialdata)
X <- microbialdata$Xenv
y <- microbialdata$Y[, order(colMeans(microbialdata$Y > 0),
                     decreasing = TRUE)[21:40]]
fit <- gllvm(y, X, formula = ~ pH + Phosp, family = poisson())
fit$logL
ordiplot(fit)
coefplot(fit)


# Inclusion of structured random row effect
sDesign<-data.frame(Site = microbialdata$Xenv$Site)
fit <- gllvm(y, X, formula = ~ pH + Phosp, family = poisson(),
            studyDesign=sDesign, row.eff=~(1|Site))

## Load a dataset from the mvabund package
```

```
library(mvabund)
data(antTraits)
y <- as.matrix(antTraits$abund)
X <- as.matrix(antTraits$env)
TR <- antTraits$traits
# Fit model with environmental covariates Bare.ground and Shrub.cover
fit <- gllvm(y, X, formula = ~ Bare.ground + Shrub.cover,
            family = poisson())
ordiplot(fit)
coefplot(fit)

## Example 1: Fit model with two unconstrained latent variables
# Using variational approximation:
fitv0 <- gllvm(y, family = "negative.binomial", method = "VA")
ordiplot(fitv0)
plot(fitv0, mfrow = c(2,2))
summary(fitv0)
confint(fitv0)

## Example 1a: Fit concurrent ordination model with two latent variables and with
# quadratic response model
# We scale and centre the  predictors to improve convergence
fity1 <- gllvm(y, X = scale(X), family = "negative.binomial",
              num.lv.c=2, method="VA")
ordiplot(fity1, biplot = TRUE)

#'## Example 1b: Fit constrained ordination model with two latent variables and with
# random canonical coefficients
fity2 <- gllvm(y, X = scale(X), family = "negative.binomial",
              num.RR=2, randomB="LV", method="VA")

# Using Laplace approximation: (this line may take about 30 sec to run)
fitl0 <- gllvm(y, family = "negative.binomial", method = "LA")
ordiplot(fitl0)

# Poisson family:
fit.p <- gllvm(y, family = poisson(), method = "LA")
ordiplot(fit.p)
# Use poisson model as a starting parameters for ZIP-model, this line
# may take few minutes to run
fit.z <- gllvm(y, family = "ZIP", method = "LA",
              control.start = list(start.fit = fit.p))
ordiplot(fit.z)


## Example 2: gllvm with environmental variables
# Fit model with two latent variables and all environmental covariates,
fitvX <- gllvm(formula = y ~ X, family = "negative.binomial")
ordiplot(fitvX, biplot = TRUE)
coefplot(fitvX)
# Fit model with environmental covariates Bare.ground and Shrub.cover
fitvX2 <- gllvm(y, X, formula = ~ Bare.ground + Shrub.cover,
 family = "negative.binomial")
```

```
ordiplot(fitvX2)
coefplot(fitvX2)
# Use 5 initial runs and pick the best one
fitvX_5 <- gllvm(y, X, formula = ~ Bare.ground + Shrub.cover,
 family = "negative.binomial", control.start=list(n.init = 5, jitter.var = 0.1))
ordiplot(fitvX_5)
coefplot(fitvX_5)

## Example 3: Data in long format
# Reshape data to long format:
datalong <- reshape(data.frame(cbind(y,X)), direction = "long",
                   varying = colnames(y), v.names = "y")
head(datalong)
fitvLong <- gllvm(data = datalong, formula = y ~ Bare.ground + Shrub.cover,
               family = "negative.binomial")

## Example 4: Fourth corner model
# Fit fourth corner model with two latent variables
fitF1 <- gllvm(y = y, X = X, TR = TR, family = "negative.binomial")
coefplot(fitF1)
# Fourth corner can be plotted also with next lines
#fourth = fitF1$fourth.corner
#library(lattice)
#a = max( abs(fourth) )
#colort = colorRampPalette(c("blue","white","red"))
#plot.4th = levelplot(t(as.matrix(fourth)), xlab = "Environmental Variables",
#              ylab = "Species traits", col.regions = colort(100),
#              at = seq( -a, a, length = 100), scales = list( x = list(rot = 45)))
#print(plot.4th)

# Specify model using formula
fitF2 <- gllvm(y = y, X = X, TR = TR,
 formula = ~ Bare.ground + Canopy.cover * (Pilosity + Webers.length),
 family = "negative.binomial")
ordiplot(fitF2)
coefplot(fitF2)

## Include species specific random slopes to the fourth corner model
fitF3 <- gllvm(y = y, X = X, TR = TR,
 formula = ~ Bare.ground + Canopy.cover * (Pilosity + Webers.length),
 family = "negative.binomial", randomX = ~ Bare.ground + Canopy.cover,
 control.start = list(n.init = 3))
ordiplot(fitF3)
coefplot(fitF3)


## Example 5: Fit Tweedie model
# Load coral data
data(tikus)
ycoral <- tikus$abund
# Let's consider only years 1981 and 1983
ycoral <- ycoral[((tikus$x$time == 81) + (tikus$x$time == 83)) > 0, ]
# Exclude species which have observed at less than 4 sites
```

```
ycoral <- ycoral[-17, (colSums(ycoral > 0) > 4)]
# Fit Tweedie model for coral data (this line may take few minutes to run)
fit.twe <- gllvm(y = ycoral, family = "tweedie", method = "LA")
ordiplot(fit.twe)

## Example 6: Random row effects
fitRand <- gllvm(y, family = "negative.binomial", row.eff = "random")
ordiplot(fitRand, biplot = TRUE)
```

logLik.gllvm *Log-likelihood of gllvm*

Description

Extracts Log-likelihood from 'gllvm' objects.

Usage

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

Arguments

object	an object of class 'gllvm'.
...	not used.

Author(s)

David I. Warton, Jenni Niku

Examples

```
## Not run:
## Load a dataset from the mvabund package
data(antTraits)
y <- as.matrix(antTraits$abund)
# Fit gllvm model
fit <- gllvm(y = y, family = poisson())
# log-Likelihood:
logLik(fit)

## End(Not run)
```

`microbialdata` *Microbial community data*

Description

Microbial community data consist of abundances of 985 bacteria species measured at 56 soil sample sites from three regions, Kilpisjarvi (Finland), Ny-Alesund (Norway), and Mayrhofen (Austria). In addition to bacteria counts, three continuous environmental variables (pH, available phosphorous and soil organic matter) were measured from each soil sample.

Usage

```
data(microbialdata)
```

Format

Y A data frame with abundances of 985 bacteria species measured at 56 soil sample sites

X Environmental variables SOM: soil organic matter, pH: soil pH value and Phosp: available phosphorus and information from the samples, including Region: sampling region (Kilpisjarvi (Finland), Ny-Alesund (Norway), and Mayrhofen (Austria).), Site: sampling site and Soiltype: soil sample type (top soil (T) or bottom soil (B))

References

Kumar, M., Brader, G., Sessitsch, A., Mäki, A., van Elsas, J.D., and Nissinen, R. (2017). Plants Assemble Species Specific Bacterial Communities from Common Core Taxa in Three Arcto-Alpine Climate Zones. Frontiers in Microbiology, 8:12.

Niku, J., Warton, D. I., Hui, F. K. C., and Taskinen, S. (2017). Generalized linear latent variable models for multivariate count and biomass data in ecology. Journal of Agricultural, Biological, and Environmental Statistics, 22:498-522.

`ordiplot.gllvm` *Plot latent variables from gllvm model*

Description

Plots latent variables and their corresponding coefficients (biplot).

Usage

```
## S3 method for class 'gllvm'
ordiplot(
  object,
  biplot = FALSE,
  ind.spp = NULL,
  alpha = 0.5,
  main = NULL,
  which.lvs = c(1, 2),
  predict.region = FALSE,
  level = 0.95,
  jitter = FALSE,
  jitter.amount = 0.2,
  s.colors = 1,
  s.cex = 1.2,
  symbols = FALSE,
  cex.spp = 0.7,
  spp.colors = "blue",
  arrow.scale = 0.8,
  arrow.spp.scale = 0.8,
  arrow.ci = TRUE,
  arrow.lty = "solid",
  spp.arrows = NULL,
  spp.arrows.lty = "dashed",
  cex.env = 0.7,
  lab.dist = 0.1,
  lwd.ellips = 0.5,
  col.ellips = 4,
  lty.ellips = 1,
  type = NULL,
  rotate = TRUE,
  ...
)
```

Arguments

`object` an object of class ’gllvm’.

`biplot` `TRUE` if both latent variables and their coefficients are plotted, `FALSE` if only latent variables.

`ind.spp` the number of response variables (usually, species) to include on the biplot. The default is none, or all if `biplot = TRUE`.

`alpha` a numeric scalar between 0 and 1 that is used to control the relative scaling of the latent variables and their coefficients, when constructing a biplot.

`main` main title.

`which.lvs` indices of two latent variables to be plotted if number of the latent variables is more than 2. A vector with length of two. Defaults to `c(1,2)`.

`predict.region` if `TRUE` or `"sites"` prediction regions for the predicted latent variables are plotted, defaults to `FALSE`. EXTENSION UNDER DEVELOPMENT: if `"species"` uncertainty estimate regions for the estimated latent variable loadings are plotted. Works only if `biplot = TRUE`.

`level` level for prediction regions.

`jitter` if `TRUE`, jittering is applied on points.

`jitter.amount` numeric, positive value indicating an amount of jittering for each point, defaults to 0.2 (jitter range).

`s.colors` colors for sites

`s.cex` size of site labels

`symbols` logical, if `TRUE` sites are plotted using symbols, if `FALSE` (default) site numbers are used

`cex.spp` size of species labels in biplot

`spp.colors` colors for sites, defaults to `"blue"`

`arrow.scale` positive value, to scale arrows

`arrow.spp.scale` positive value, to scale arrows of species

`arrow.ci` represent statistical uncertainty for arrows in constrained or concurrent ordination using confidence or prediction interval? Defaults to `TRUE`

`arrow.lty` linetype for arrows in constrained

`spp.arrows` plot species scores as arrows if outside of the range of the plot? Defaults to `FALSE` for linear response models and `TRUE` for quadratic response models.

`spp.arrows.lty` linetype for species arrows

`cex.env` size of labels for arrows in constrained ordination

`lab.dist` distance between label and arrow heads. Value between 0 and 1

`lwd.ellips` line width for prediction ellipses. See graphical parameter lwd.

`col.ellips` colors for prediction ellipses.

`lty.ellips` line type for prediction ellipses. See graphical parameter lty.

`type` which type of ordination plot to construct. Options are "residual", "conditional", and "marginal". Defaults to "residual" for GLLVMs with unconstrained latent variables and "conditional" otherwise.

`rotate` logical, if `TRUE` (default) latent variables are rotated to their principal direction using singular value decomposition

`...` additional graphical arguments.

Details

Function constructs a scatter plot of two latent variables, i.e. an ordination plot. Latent variables are re-rotated to their principal direction using singular value decomposition, so that the first plotted latent variable does not have to be the first latent variable in the model. If only one latent variable is in the fitted model, latent variables are plotted against their corresponding row indices. The latent variables are labeled using the row index of the response matrix y.

Coefficients related to latent variables are plotted in the same figure with the latent variables if `biplot = TRUE`. They are labeled using the column names of y. The number of latent variable coefficients to be plotted can be controlled by ind.spp. An argument alpha is used to control the relative scaling of the latent variables and their coefficients. If `alpha = 0.5`, the latent variables and their coefficients are on the same scale. For details for constructing a biplot, see Gabriel (1971).

For a quadratic response model, species optima are plotted. Any species scores that are outside the range of the predicted site scores are not directly plotted, but their main direction is indicated with arrows instead. This ensures that the plot remains on a reasonable scale.

Effects of environmental variables in constrained ordination are indicated with arrows. If any of the arrows exceeds the range of the plot, arrows are scaled to 80 but so that the relative contribution of predictors is maintained. If standard errors are available in the provided model, the slopes of environmental variables for which the 95 are slightly less intensely coloured.

For constrained ordination, a conditional plot includes both fixed- and random-effects to optimally represent species co-occurrence patterns, corresponding to "conditional" site scores in `getLV.gllvm`. Marginal corresponds to an ordination plot that excludes residual patterns (i.e. excluding the random-effect), so that it is only available with num.lv.c>0 or num.RR>0. A conditional plot requires num.lv.c>0. The "residual" type corresponds to an ordination diagram of only residual patterns. See getLV.gllvm for details.

Note

- If error is occurred when using `ordiplot()`, try full name of the function `ordiplot.gllvm()` as functions named 'ordiplot' might be found in other packages as well.

Author(s)

Jenni Niku <jenni.m.e.niku@jyu.fi>, Francis K.C. Hui, Bert van der Veen

References

Gabriel, K. R. (1971). The biplot graphic display of matrices with application to principal component analysis. Biometrika, 58, 453-467.

See Also

`getLV.gllvm`.

Examples

```
#'# Extract subset of the microbial data to be used as an example
data(microbialdata)
y <- microbialdata$Y[, order(colMeans(microbialdata$Y > 0),
                     decreasing = TRUE)[21:40]]
fit <- gllvm(y, family = poisson())
fit$logL
ordiplot(fit, predict.region = TRUE)
## Not run:
#'## Load a dataset from the mvabund package
data(antTraits)
y <- as.matrix(antTraits$abund)
```

```
fit <- gllvm(y, family = poisson())
# Ordination plot:
ordiplot(fit)
# Biplot with 10 species
ordiplot(fit, biplot = TRUE, ind.spp = 10)

## End(Not run)
```

plot.gllvm *Plot Diagnostics for an gllvm Object*

Description

Five plots (selectable by which) are currently available: a plot of residuals against linear predictors of fitted values, a Normal Q-Q plot of residuals with a simulated point-wise 95% confidence interval envelope, residuals against row index and column index and scale location plot.

Usage

```
## S3 method for class 'gllvm'
plot(
  x,
  which = 1:5,
  caption = c("Residuals vs linear predictors", "Normal Q-Q", "Residuals vs row",
    "Residuals vs column", "Scale-Location"),
  var.colors = NULL,
  add.smooth = TRUE,
  envelopes = TRUE,
  reps = 150,
  envelope.col = c("blue", "lightblue"),
  n.plot = NULL,
  ...
)
```

Arguments

`x` an object of class 'gllvm'.

`which` if a subset of the plots is required, specify a subset of the numbers 1:5, see caption below.

`caption` captions to appear above the plots.

`var.colors` colors for responses, vector with length of number of response variables or 1. Defaults to NULL, when different responses have different colors.

`add.smooth` logical indicating if a smoother should be added.

`envelopes` logical, indicating if simulated point-wise confidence interval envelope will be added to Q-Q plot, defaults to `TRUE` `reps` number of replications when simulating confidence envelopes for normal Q-Q plot

`envelope.col` colors for envelopes, vector with length of two

`n.plot` number of species (response variables) to be plotted. Defaults to `NULL` when all response variables are plotted. Might be useful when data is very high dimensional.

`...` additional graphical arguments.

Details

plot.gllvm is used for model diagnostics. Dunn-Smyth residuals (randomized quantile residuals) (Dunn and Smyth, 1996) are used in plots. Colors indicate different species.

Author(s)

Jenni Niku <jenni.m.e.niku@jyu.fi>

References

Dunn, P. K., and Smyth, G. K. (1996). Randomized quantile residuals. Journal of Computational and Graphical Statistics, 5, 236-244.

Hui, F. K. C., Taskinen, S., Pledger, S., Foster, S. D., and Warton, D. I. (2015). Model-based approaches to unconstrained ordination. Methods in Ecology and Evolution, 6:399-411.

See Also

`gllvm`, `residuals.gllvm`

Examples

```
## Not run:
## Load a dataset from the mvabund package
data(antTraits)
y <- as.matrix(antTraits$abund)
# Fit gllvm model with Poisson family
fit <- gllvm(y, family = poisson())
# Plot residuals
plot(fit, mfrow = c(3,2))

\donttest{
# Fit gllvm model with negative binomial family
fitnb <- gllvm(y = y, family = "negative.binomial")
# Plot residuals
plot(fitnb, mfrow = c(3,2))
# Plot only two first plots
plot(fitnb, which = 1:2, mfrow = c(1,2))
}

## End(Not run)
```

`predict.gllvm` *Predict Method for gllvm Fits*

Description

Obtains predictions from a fitted generalized linear latent variable model object.

Usage

```
## S3 method for class 'gllvm'
predict(
  object,
  newX = NULL,
  newTR = NULL,
  newLV = NULL,
  type = "link",
  level = 1,
  offset = TRUE,
  ...
)
```

Arguments

`object` an object of class ’gllvm’.

`newX` A new data frame of environmental variables. If omitted, the original matrix of environmental variables is used.

`newTR` A new data frame of traits for each response taxon. If omitted, the original matrix of traits is used.

`newLV` A new matrix of latent variables. If omitted, the original matrix of latent variables is used. Note that number of rows/sites must be the same for `newX` (if X covariates are included in the model).

`type` the type of prediction required. The default (`"link"`) is on the scale of the linear predictors; the alternative `"response"` is on the scale of the response variable. that is, the predictions for the binomial model are predicted probabilities. In case of ordinal data, `type = "response"` gives predicted probabilities for each level of ordinal variable.

`level` specification for how to predict. Level one (`level = 1`) attempts to use the predicted site scores from variational approximations or laplace approximation or given site scores in `newLV`. Level 0 sets the latent variable to zero. Defaults to 1.

`offset` specification whether of not offset values are included to the predictions in case they are in the model, defaults to `TRUE` when offset values that are used to fit the model are included to the predictions. Alternatives are matrix/vector (number of rows must match with the `newX`) of new offset values or `FALSE`, when offsets are ignored.

`...` not used.

Details

If newX, newTR and newLV are omitted the predictions are based on the data used for fitting the model. Notice that newTR need to match with the number of species in the original data. Instead, new sites can be specified in newX. If predictors newX (and newTR) are given, and newLV is not, latent variables are not used in the predictions.

Value

A matrix containing requested predictor types.

Author(s)

Jenni Niku <jenni.m.e.niku@jyu.fi>, David Warton

Examples

```
# Load a dataset from the mvabund package
data(antTraits)
y <- as.matrix(antTraits$abund)
X <- scale(antTraits$env[, 1:3])
# Fit gllvm model
fit <- gllvm(y = y, X, family = poisson())
# fitted values
predfit <- predict(fit, type = "response")

# linear predictors
predlin <- predict(fit)
# Predict new sites:
# Generate matrix of environmental variables for 10 new sites
xnew <- cbind(rnorm(10), rnorm(10), rnorm(10))
colnames(xnew) <- colnames(X)
predfit <- predict(fit, newX = xnew, type = "response", level = 0)

TR <- (antTraits$tr[, 1:3])
fitt <- gllvm(y = y, X, TR, family = poisson())
# linear predictors
predlin <- predict(fitt)
# Predict new sites:
# Generate matrix of environmental variables for 10 new sites
xnew <- cbind(rnorm(10), rnorm(10), rnorm(10))
colnames(xnew) <- colnames(X)
# Generate matrix of traits for species
trnew <- data.frame(Femur.length = rnorm(41), No.spines = rnorm(41),
 Pilosity = factor(sample(0:3, 41, replace = TRUE)))
predfit <- predict(fitt, newX = xnew, newTR = trnew, type = "response", level = 0)
```

predictLVs.gllvm *Predict latent variables for gllvm Fits*

Description

Obtains predictions for latent variables from a fitted generalized linear latent variable model object. Currently works only for the variational approximation method.

Usage

```
## S3 method for class 'gllvm'
predictLVs(object, newX = NULL, newY = object$y, ...)
```

Arguments

object	an object of class 'gllvm'.
newX	A new data frame of environmental variables. If omitted, the original matrix of environmental variables is used.
newY	A new response data. Defaults to the dataset used for original model fit.
...	not used.

Details

Obtains predictions for latent variables from a fitted generalized linear latent variable model object.

Value

A matrix containing requested predictor types.

Author(s)

David Warton, Jenni Niku <jenni.m.e.niku@jyu.fi>

Examples

```
# Load a dataset from the mvabund package
data(antTraits)
y <- as.matrix(antTraits$abund)
X <- scale(antTraits$env[, 1:3])
# Fit gllvm model
fit <- gllvm(y = y, X, family = poisson())
# fitted values
predLVs <- predictLVs.gllvm(fit)
```

randomCoefplot.gllvm *Plot random slope coefficients*

Description

Plots random slopes and their prediction intervals.

Usage

```
## S3 method for class 'gllvm'
randomCoefplot(
  object,
  y.label = TRUE,
  which.Xcoef = NULL,
  cex.ylab = 0.5,
  mfrow = NULL,
  mar = c(4, 6, 2, 1),
  xlim.list = NULL,
  order = FALSE,
  ...
)
```

Arguments

`object`	an object of class 'gllvm'.
`y.label`	logical, if `TRUE` (default) colnames of y with respect to coefficients are added to plot.
`which.Xcoef`	fector indicating which covariate coefficients will be plotted. Can be vector of covariate names or numbers. Default is NULL when all covariate coefficients are plotted.
`cex.ylab`	the magnification to be used for axis annotation relative to the current setting of cex.
`mfrow`	same as `mfrow` in `par`. If `NULL` (default) it is determined automatically.
`mar`	vector of length 4, which defines the margin sizes: `c(bottom, left, top, right)`. Defaults to `c(4,5,2,1)`.
`xlim.list`	list of vectors with length of two to define the intervals for x axis in each covariate plot. Defaults to NULL when the interval is defined by the range of point estimates and confidence intervals
`order`	logical, if `TRUE` (default), coefficients are sorted according to the point estimates
`...`	additional graphical arguments.

Author(s)

Jenni Niku <jenni.m.e.niku@jyu.fi>, Francis K.C. Hui, Bert van der Veen, Sara Taskinen,

Examples

```
## Not run:
## Load a dataset from the mvabund package
data(antTraits)
y <- as.matrix(antTraits$abund)
X <- as.matrix(antTraits$env)
TR <- antTraits$traits
# Fit model with random slopes
fitF <- gllvm(y = y, X = X, TR = TR,
 formula = ~ Bare.ground + Bare.ground : Webers.length,
 family = poisson(), randomX = ~ Bare.ground)
randomCoefplot(fitF)

## End(Not run)
```

---

residuals.gllvm *Dunn-Smyth residuals for gllvm model*

Description

Calculates Dunn-Smyth residuals for gllvm model.

Usage

```
## S3 method for class 'gllvm'
residuals(object, ...)
```

Arguments

`object` an object of class ’gllvm’.

`...` not used.

Details

Computes Dunn-Smyth residuals (randomized quantile residuals, Dunn and Smyth, 1996) for gllvm model. For the observation Y_{ij} Dunn-Smyth residuals are defined as

$$r_{ij} = \Phi^{-1}(u_{ij}F_{ij}(y_{ij}) + (1 - u_{ij})F_{ij}^{-}(y_{ij})),$$

where $\Phi(.)$ and $F_{ij}(.)$ are the cumulative probability functions of the standard normal distribution, $F_{ij}^{-}(y))$ is the limit as $F_{ij}(y)$ is approached from the negative side, and u_{ij} has been generated at random from the standard uniform distribution.

Value

`residuals` matrix of residuals

`linpred` matrix of linear predictors

Author(s)

Jenni Niku <jenni.m.e.niku@jyu.fi>

References

Dunn, P. K., and Smyth, G. K. (1996). Randomized quantile residuals. Journal of Computational and Graphical Statistics, 5, 236-244.

Hui, F. K. C., Taskinen, S., Pledger, S., Foster, S. D., and Warton, D. I. (2015). Model-based approaches to unconstrained ordination. Methods in Ecology and Evolution, 6:399-411.

Examples

```
## Not run:
# Load a dataset from the mvabund package
data(antTraits)
y <- as.matrix(antTraits$abund)
# Fit gllvm model
fit <- gllvm(y = y, family = poisson())
# residuals
res <- residuals(fit)

## End(Not run)
```

se.gllvm *Standard errors for gllvm model*

Description

Calculates Hessian and standard errors for gllvm model.

Usage

```
## S3 method for class 'gllvm'
se(object, ...)
```

Arguments

`object`	an object of class 'gllvm'.
`...`	not used.

Details

Computes Hessian and standard errors for gllvm model.

Value

`sd`	list of standard errors of parameters
`Hess`	list including Hessian matrix and approximative covariance matrix of parameters

Author(s)

Jenni Niku <jenni.m.e.niku@jyu.fi>

References

Dunn, P. K., and Smyth, G. K. (1996). Randomized quantile residuals. Journal of Computational and Graphical Statistics, 5, 236-244.

Hui, F. K. C., Taskinen, S., Pledger, S., Foster, S. D., and Warton, D. I. (2015). Model-based approaches to unconstrained ordination. Methods in Ecology and Evolution, 6:399-411.

Examples

```
data(spider)
mod <- gllvm(spider$abund, num.lv = 2, family = "poisson", sd.errors = FALSE)
# Calculate standard errors after fitting
sdErr <- se(mod)
# Store the standard errors in the right place
mod$sd <-sdErr$sd
# Store the Hessian in the right place
mod$Hess <- sdErr$Hess
```

`simulate.gllvm` *Simulate data from gllvm fit*

Description

Generate new data using the fitted values of the parameters

Usage

```
## S3 method for class 'gllvm'
simulate(object, nsim = 1, seed = NULL, conditional = FALSE, ...)
```

Arguments

`object`	an object of class 'gllvm'.
`nsim`	an optional positive integer specifying the number of simulated datasets. Defaults to 1.
`seed`	an optional integer to set seed number, passed to set.seed. Defaults to a random seed number.
`conditional`	if `conditional` = `FALSE` simulates marginally over the latent variables.
`...`	not used.

Details

simulate function for gllvm objects.

Value

A matrix containing generated data.

Author(s)

David Warton, Jenni Niku <jenni.m.e.niku@jyu.fi>

Examples

```
# Load a dataset from the mvabund package
data(antTraits)
y <- as.matrix(antTraits$abund)
X <- scale(antTraits$env[, 1:3])
# Fit gllvm model
fit <- gllvm(y = y, X, family = poisson())
# Simulate data
newdata <- simulate(fit)
```

Skabbholmen *Skabbholmen island data*

Description

Dataset of ordinal observations of plants, on the island Skabbholmen in the Stocholm archipelago. Includes 65 unique sites and 70 species, surveyed in two different years.

Usage

```
data(Skabbholmen)
```

Format

Y A data frame with ordinal of 70 plant species measured at 126 plots.

X A matrix of 2 predictor variables at 126 plots.

species A matrix of full species names and abbreviations used in the community data (Y).

Details

Observations of vascular plant cover in 126 one-square-meter plots divided over four transects. The ordinal responses are on a five-degree Hult-Sernander-Du Rietz scale, and were originally recorded by Wolfgang and Cramer (1987) and additionally analyzed by ter Braak (1987). There is a total of 64 unique sites, that were surveyed in two different years (1978 and 1984), but two plots were only surveyed in one year (thus bringing the total number of rows in the data to 126). The plots were located on an elevation gradient, running from the shoreline to the edge of old-growth forest. Elevation to the shoreline was recorded in centimeters during the sampling in 1978.

This dataset was published with permission from the CANOCO FORTRAN package example datasets.

References

ter Braak, C.J.F. and Smilauer, P. (1998). CANOCO reference manual and user's guide to CANOCO for Windows: software for canonical community ordination (version 4). Microcomputer Power, New York, New York, USA.

Jongman, E., & Jongman, S. R. R. (1995). Data analysis in community and landscape ecology. Cambridge university press.

ter Braak, C.J.F. (1987). The analysis of vegetation-environment relationships by canonical correspondence analysis. Vegetatio, 69(1), 69-77.

Cramer, W. & Hytteborn, H. (1987). The separation of fluctuation and long-term change in vegetation dynamics of a rising seashore. Vegetatio, 69, 157–167.

Examples

```
# Uncomment the example
#data(Skabbholmen)
#Y <- Skabbholmen$Y
#X <- Skabbholmen$X
#model <- gllvm(y = Y, X = X,
#    num.RR = 2,
#    family = "ordinal",
#    zeta.struc="common",
#    row.eff=~(1|transectID))
```

---

summary.gllvm *Summarizing gllvm model fits*

Description

A summary of the fitted 'gllvm' object, including function call, distribution family and model parameters.

Usage

```
## S3 method for class 'gllvm'
summary(
  object,
  by = "all",
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  dispersion = FALSE,
  spp.intercepts = FALSE,
  row.intercepts = FALSE,
  Lvcoefs = FALSE,
  rotate = TRUE,
  type = NULL,
  ...
)

## S3 method for class 'summary.gllvm'
print(x, ...)
```

Arguments

object an object of class 'gllvm'

by By = "all" (default) will return a Wald statistics per predictor and LV if the ordination includes predictors, by = "terms" will return a multivariate Wald statistic per predictor (displayed at first LV), and by = "LV" will do the same but per dimension (displayed at first predictors).

digits the number of significant digits to use when printing

signif.stars If TRUE, significance stars are printed for each coefficient, defaults to TRUE

dispersion option to return dispersion parameters, defaults to FALSE

spp.intercepts option to return species intercepts, defaults to FALSE

row.intercepts option to return row intercepts, defaults to FALSE

Lvcoefs option to return species scores in the ordination, defaults to FALSE. Returns species optima for quadratic model.

rotate defaults to TRUE. If TRUE rotates the output of the latent variables to principal direction, so that it coincides with the ordiplot results. If both unconstrained and constrained latent variables are included, predictor slopes are not rotated.

type to match "type" in ordiplot.gllvm

... not used.

x a summary object

Details

Various options are available to include extra parameter estimates in the summary, which have been excluded by default, for readability.

Author(s)

Jenni Niku <jenni.m.e.niku@jyu.fi>, Bert van der Veen

Examples

```
## Not run:
## Load a dataset from the mvabund package
data(antTraits)
y <- as.matrix(antTraits$abund)
# Fit gllvm model
fit <- gllvm(y = y, family = poisson())
summary(fit)

## End(Not run)
```

vcov.gllvm *Returns variance-covariance matrix of coefficients in a GLLVM.*

Description

Returns the variance-covariance matrix of the parameters from a GLLVM. If the variance-covariance matrix was not calculated after model fitting, this function will have to calculate the variance-covariance matrix, which may be computational intensive for a large number of species and/or sites.

Usage

```
## S3 method for class 'gllvm'
vcov(object, ...)
```

Arguments

`object`	an object of class 'gllvm'.
`...`	not used.

Details

Calculates the variance-covariance matrix of a GLLVM object using `se.gllvm`, which may be computational intensive with many parameters.The parameters might have unintuitive names. Fixed-effects coefficients are labeled "b", and are ordered per species as: 1) intercepts 2) fixed-effects slopes. Coefficients of the latent variables are labled "lambda" (linear coefficients) or "lambda2".

Author(s)

Bert van der Veen

Index