Package ‘vinereg’

February 9, 2024

Type Package
Title D-Vine Quantile Regression
Version 0.10.0
Maintainer Thomas Nagler <mail@tnagler.com>
License GPL-3
Imports rvinecopulib (>= 0.6.1.1.2), kde1d (>= 1.0.2), Rcpp, assertthat
LinkingTo rvinecopulib, RcppEigen, Rcpp, BH, wdm, RcppThread, kde1d
RoxygenNote 7.2.3
Suggests knitr, rmarkdown, ggplot2, AppliedPredictiveModeling, quantreg, tidyr, dplyr, purrr, scales, mgcv, testthat, covr
VignetteBuilder knitr
URL https://tnagler.github.io/vinereg/
BugReports https://github.com/tnagler/vinereg/issues
Encoding UTF-8
NeedsCompilation yes
Author Thomas Nagler [aut, cre], Dani Kraus [ctb]
Repository CRAN
Date/Publication 2024-02-09 16:00:02 UTC

R topics documented:
cll ................................................................. 2
cpdf ............................................................. 3
cll  .......................................................... 3
plot_effects .................................................. 4
predict.vinereg .............................................. 5
vinereg ....................................................... 6

Index

cll  Conditional log-likelihood

Description

Calculates the conditional log-likelihood of the response given the covariates.

Usage

cll(object, newdata, cores = 1)

Arguments

object  an object of class vinereg.
newdata matrix of response and covariate values for which to compute the conditional
distribution.
cores  integer; the number of cores to use for computations.

Examples

# simulate data
x <- matrix(rnorm(200), 100, 2)
y <- x %*% c(1, -2)
dat <- data.frame(y = y, x = x, z = as.factor(rbinom(100, 2, 0.5)))

# fit vine regression model
fit <- vinereg(y ~ ., dat)
cll(fit, dat)
fit$stats$cll
**cpdf**  
*Conditional PDF*

**Description**  
Calculates the conditional density of the response given the covariates.

**Usage**  
```r
cpdf(object, newdata, cores = 1)
```

**Arguments**
- **object**: an object of class `vinereg`.
- **newdata**: matrix of response and covariate values for which to compute the conditional density.
- **cores**: integer; the number of cores to use for computations.

**Examples**
```r
# simulate data
x <- matrix(rnorm(200), 100, 2)
y <- x %*% c(1, -2)
dat <- data.frame(y = y, x = x, z = as.factor(rbinom(100, 2, 0.5)))

# fit vine regression model
fit <- vinereg(y ~ ., dat)

cpdf(fit, dat)
```

**cpit**  
*Conditional probability integral transform*

**Description**  
Calculates the conditional distribution of the response given the covariates.

**Usage**  
```r
cpit(object, newdata, cores = 1)
```
Arguments

object an object of class vinereg.
newdata matrix of response and covariate values for which to compute the conditional distribution.
cores integer; the number of cores to use for computations.

Examples

# simulate data
x <- matrix(rnorm(200), 100, 2)
y <- x %*% c(1, -2)
dat <- data.frame(y = y, x = x, z = as.factor(rbinom(100, 2, 0.5)))

# fit vine regression model
fit <- vinereg(y ~ ., dat)

hist(cpit(fit, dat)) # should be approximately uniform

plot_effects

Plot marginal effects of a D-vine regression model

Description

The marginal effects of a variable is the expected effect, where expectation is meant with respect to all other variables.

Usage

plot_effects(object, alpha = c(0.1, 0.5, 0.9), vars = object$order)

Arguments

object a vinereg object
alpha vector of quantile levels.
vars vector of variable names.

Examples

# simulate data
x <- matrix(rnorm(200), 100, 2)
y <- x %*% c(1, -2)
dat <- data.frame(y = y, x = x, z = as.factor(rbinom(100, 2, 0.5)))

# fit vine regression model
fit <- vinereg(y ~ ., dat)
plot_effects(fit)
**predict.vinereg**  
*Predict conditional mean and quantiles from a D-vine regression model*

**Description**  
Predict conditional mean and quantiles from a D-vine regression model

**Usage**
```r  
## S3 method for class 'vinereg'
predict(object, newdata, alpha = 0.5, cores = 1, ...)
## S3 method for class 'vinereg'
fitted(object, alpha = 0.5, ...)
```

**Arguments**
- `object`: an object of class `vinereg`.
- `newdata`: matrix of covariate values for which to predict the quantile.
- `alpha`: vector of quantile levels; NA predicts the mean based on an average of the 1:10 /11-quantiles.
- `cores`: integer; the number of cores to use for computations.
- `...`: unused.

**Value**
A data.frame of quantiles where each column corresponds to one value of `alpha`.

**See Also**
- `vinereg`

**Examples**
```r  
# simulate data  
x <- matrix(rnorm(200), 100, 2)  
y <- x %*% c(1, -2)  
dat <- data.frame(y = y, x = x, z = as.factor(rbinom(100, 2, 0.5)))

# fit vine regression model  
(fit <- vinereg(y ~ ., dat))

# inspect model  
summary(fit)  
plot_effects(fit)
```
# model predictions
mu_hat <- predict(fit, newdata = dat, alpha = NA) # mean
med_hat <- predict(fit, newdata = dat, alpha = 0.5) # median

# observed vs predicted
plot(cbind(y, mu_hat))

## fixed variable order (no selection)
(fit <- vinereg(y ~ ., dat, order = c("x.2", "x.1", "z.1")))

---

**Description**

Sequential estimation of a regression D-vine for the purpose of quantile prediction as described in Kraus and Czado (2017).

**Usage**

```r
vinereg(
  formula,
  data,
  family_set = "parametric",
  selcrit = "aic",
  order = NA,
  par_1d = list(),
  weights = numeric(),
  cores = 1,
  ...
)
```

**Arguments**

- `formula`: an object of class "formula"; same as `lm()`.
- `data`: data frame (or object coercible by `as.data.frame()`) containing the variables in the model.
- `family_set`: see `family_set` argument of `rvinecopulib::bicop()`.
- `selcrit`: selection criterion based on conditional log-likelihood. "loglik" (default) imposes no correction; other choices are "aic" and "bic".
- `order`: the order of covariates in the D-vine, provided as vector of variable names (after calling `vinereg:::expand_factors(model.frame(formula, data)))`; selected automatically if `order = NA` (default).
- `par_1d`: list of options passed to `kde1d::kde1d()`, must be one value for each margin, e.g. `list(xmin = c(0, 0, NaN))` if the response and first covariate have non-negative support.
weights optional vector of weights for each observation.
cores integer; the number of cores to use for computations.
... further arguments passed to \texttt{rvinecopulib::bicop()}.
uscale if \texttt{TRUE}, \texttt{vinereg} assumes that marginal distributions have been taken care of in a preliminary step.

Details
If discrete variables are declared as \texttt{ordered()} or \texttt{factor()}, they are handled as described in Panagiotelis et al. (2012). This is different from previous version where the data was jittered before fitting.

Value
An object of class \texttt{vinereg}. It is a list containing the elements

- \textbf{formula} the formula used for the fit.
- \textbf{selcrit} criterion used for variable selection.
- \textbf{model_frame} the data used to fit the regression model.
- \textbf{margins} list of marginal models fitted by \texttt{kde1d::kde1d()}.
- \textbf{vine} an \texttt{rvinecopulib::vinecop_dist()} object containing the fitted D-vine.
- \textbf{stats} fit statistics such as conditional log-likelihood/AIC/BIC and p-values for each variable's contribution.
- \textbf{order} order of the covariates chosen by the variable selection algorithm.
- \textbf{selected_vars} indices of selected variables.

Use \texttt{predict.vinereg()} to predict conditional quantiles. \texttt{summary.vinereg()} shows the contribution of each selected variable with the associated p-value derived from a likelihood ratio test.

References
Kraus and Czado (2017), D-vine copula based quantile regression, Computational Statistics and Data Analysis, 110, 1-18

See Also
\texttt{predict.vinereg}

Examples
\begin{verbatim}
# simulate data
x <- matrix(rnorm(200), 100, 2)
y <- x %*% c(1, -2)
dat <- data.frame(y = y, x = x, z = as.factor(rbinom(100, 2, 0.5)))
\end{verbatim}
# fit vine regression model
(fit <- vinereg(y ~ ., dat))

# inspect model
summary(fit)
plot_effects(fit)

# model predictions
mu_hat <- predict(fit, newdata = dat, alpha = NA) # mean
med_hat <- predict(fit, newdata = dat, alpha = 0.5) # median

# observed vs predicted
plot(cbind(y, mu_hat))

## fixed variable order (no selection)
(fit <- vinereg(y ~ ., dat, order = c("x.2", "x.1", "z.1")))
Index

as.data.frame(), 6

cll, 2
cpdf, 3
cpit, 3

fitted.vinereg(predict.vinereg), 5

cde1d::kde1d(), 6, 7

lm(), 6

plot_effects, 4
predict.vinereg, 5, 7
predict.vinereg(), 7

rvinecopulib::bicop(), 6, 7
rvinecopulib::vinereg_dist(), 7

vinereg, 5, 6