Package 'qrjoint'

March 16, 2016

Version 1.0-0

Date 2016-03-16

Title Joint Estimation in Linear Quantile Regression

Author Surya Tokdar <tokdar@stat.duke.edu>

Maintainer Surya Tokdar <tokdar@stat.duke.edu>

Depends R (>= 2.6), stats, graphics, grDevices, quantreg

Imports splines, coda, Matrix, kernlab

Description
Joint estimation of quantile specific intercept and slope parameters in a linear regression setting.

License GPL-2

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-03-16 18:37:04

R topics documented:

`chull.center` *Fast Interior Point Center of Multivariate Data*

Description

Calculates an interior point by averaging a small number of near-extreme points of the cloud.

Usage

```
chull.center(x, maxEPts = ncol(x) + 1, plot = FALSE)
```

Arguments

`x`	a matrix giving the data cloud.
`maxEPts`	integer giving the maximum number of (near)-extreme points to be used in averaging. Default is `ncol(x)+1`.
`plot`	logical indicating whether a pairwise scatter plot should be made

Details

Near extreme points are found in a space-filling manner by adding points with minimum residual conditional variance given points already included under a smooth GP specification. See Yang and Tokdar (2015), Section B.1. for more details.

Value

Returns an interior point of the data cloud. The positions of the near extreme points are returned as the attribute `"EPts"`.

Examples

```
p <- 9
n <- 200
u <- runif(n)
require(splines)
x <- bs(u, df = p)
chull.center(x, plot = TRUE)
```

coef.qrjoint *Regression Coefficient Extraction from qrjoint Model Fit*

Description

Post process MCMC output from qrjoint to create summaries of intercept and slope function estimates

Usage

```
## S3 method for class 'qrjoint'
coef(object, burn.perc = 0.5, nmc = 200, plot = FALSE, show.intercept = TRUE,
        reduce = TRUE, ...)
```

Arguments

object	a fitted model of the class qrjoint.
burn.perc	a positive fraction indicating what fraction of the saved draws are to be discarded as burn-in
nmc	integer giving the number of samples, post burn-in, to be used in Monte Carlo averaging
plot	logical indicating if plots are to be made
show.intercept	whether to plot the intercept curve when plot = TRUE
reduce	logical indicating if the tail-expanded grid of tau values is to be reduced to the regular increment grid
...	limited plotting parameters that are passed onto the call of getBands

Value

Extracts posterior draws of intercept and slope functions from saved draws of model parameters. A plot may be obtained if plot = TRUE. A list is returned invisibly with two fields.

beta.samp	a matrix with nmc many columns and (p+1)*length(tau.grid) many rows.
beta.est	a list of length (p+1), j-th element giving a 3-column matrix of median, 2.5th and 97.5th percentiles of the posterior distribution of β_j

See Also

qrjoint and summary.qrjoint for model fitting under qrjoint. Also see getBands for plotting credible bands for coefficients.

Examples

```
## Plasma data analysis

# recoding variables
data(plasma)
plasma$Sex <- as.factor(plasma$Sex)
plasma$SmokStat <- as.factor(plasma$SmokStat)
plasma$VitUse <- 3 - plasma$VitUse
plasma$VitUse <- as.factor(plasma$VitUse)

# creating predictors and response (beta carotene concentration in the plasma)
X <- model.matrix(BetaPlasma ~ Age + Sex + SmokStat + Quetelet + VitUse + Calories +
        Fat + Fiber + Alcohol + Cholesterol + BetaDiet, data = plasma)[,-1]
Y <- plasma$BetaPlasma

# model fitting with 50 posterior samples from 100 iterations (thin = 2)
fit.qrj <- qrjoint(X, Y, 50, 2)

## Not run:
betas <- coef(fit.qrj) ## no plots

## End(Not run)
betas <- coef(fit.qrj, plot = TRUE, col = "darkgreen") ## estimates are plotted
```

---

getBands *Posterior Credible Bands*

Description

Calculate and display credible bands of a vector of parameters from a sample of draws. Most suitable when the vector represents a discretized version of a function.

Usage

```
getBands(b, col = 2, lwd = 1, plot = TRUE, add = FALSE,
x = seq(0,1,len=nrow(b)), remove.edges = TRUE, ...)
```

Arguments

b	a matrix of sampled draws of a vector, columns giving samples and rows giving elements of the vector
col	color of the median line and 95% bands, usual color codes could be used
lwd	line width for the median line
plot	logical indicating whether plots are to be drawn, default is TRUE
add	logical indicating whether plot is to be added to existing plot, default is FALSE

x	indexing the parameter coordinates. When b represents a (discretized) function, x can be taken as the function argument values. Needed when plot is to be created. Default sets it to a uniform grid of points over [0,1].
remove.edges	logical indicating whether the first and last entries of b are to be removed from plotting. This is helpful in qrjoint plots, where the two extremes could be Inf or nearly Inf.
...	limited number of plotting parameters

Value

returns median, 2.5th and 97.5th percentiles as a 3-column matrix.

See Also

coef.qrjoint

Examples

```
## toy example

x <- 2*pi*seq(0,1,.01)
fsamp <- replicate(100, rnorm(1,0,0.1) + rnorm(1,1,0.2) * cos(x))
getBands(fsamp)
getBands(fsamp, x = x, col = 3, remove.edges = FALSE, xlab = "x", ylab = "f", bty = "n")
getBands(fsamp, x = x, col = "darkgreen", remove.edges = FALSE, xlab = "x", ylab = "f")
```

---

plasma *Plasma Concentration of Beta-Carotene and Retinol*

Description

Plasma concentration levels of beta-carotene and retinol along with dietary intake and drug usage measurements for 315 patients who had an elective surgical procedure during a three-year period to biopsy or remove a lesion of the lung, colon, breast, skin, ovary or uterus that was found to be non-cancerous.

Usage

```
data(plasma)
```

Format

A data frame with 315 observations on the following 14 variables.

Age age (years)

Sex sex (1=Male, 2=Female)

SmokStat smoking status (1=Never, 2=Former, 3=Current)

`Quetelet` Quetlet index, aka, BMI (weight / height^2)

`VitUse` vitamin use (`0`=No, `1`=Yes, not often, `2`=Yes, fairly often)

`Calories` number of calories consumed per day

`Fat` grams of fat consumed per day

`Fiber` grams of fiber consumed per day

`Alcohol` number of alcoholic drinks consumed per week

`Cholesterol` cholesterol consumed (mg per day)

`BetaDiet` dietary beta-carotene consumed (mcg per day)

`RetDiet` dietary retinol consumed (mcg per day)

`BetaPlasma` plasma beta-carotene concentration (ng/ml)

`RetPlasma` plasma retinol concentration (ng/ml)

Details

Dietary intakes are self-reported. Results from analyzing this data should be used with caution!

Source

Statlib database

References

Nierenberg, D. W., T. A. Stukel, J. A. Baron, B. J. Dain, and E. R. Greenberg (1989). Determinants of plasma levels of beta-carotene and retinol. *American Journal of Epidemiology*, 130(3), 511–521.

Examples

```
data(plasma)
```

---

qrjoint *Joint Estimation of Linear Quantile Planes*

Description

Estimate intercept and slope functions within a joint linear regression model of the quantiles, with possible right censoring of the response.

Usage

```
qrjoint(x, y, nsamp = 1e3, thin = 10, cens = rep(0,length(y)),
    incr = 0.01, par = "prior", nknots = 6,
    hyper = list(sig = c(.1,.1), lam = c(6,4), kap = c(.1,.1,1)),
    shrink = FALSE, prox.range = c(.2,.95), acpt.target = 0.15,
    ref.size = 3, blocking = "std5", temp = 1, expo = 2,
    blocks.mu, blocks.S, fix.nu=FALSE)

## S3 method for class 'qrjoint'
update(object, nadd, append = TRUE, ...)
```

Arguments

x	numeric design matrix of explanatory variables (do not include a column of ones). Could be given as a data frame. Rows are observations and columns are variables. Could be given as a dimensionless vector when there is a single predictor.
y	numeric vector of response data.
nsamp	number of posterior samples to be saved; defaults to 1000.
thin	thinning rate for the Markov chain sampler – one posterior sample is saved per thin iterations. Defaults to 10. The Markov chain sampler runs for a total of nsamp * thin many iterations.
cens	censoring status of response. Must be a vector of 0s and 1s of length same as length(y), with 1 indicating right censoring, and, 0 indicating no censoring. Defaults to all zeros.
incr	tau grid increment. Defaults to 0.01.
par	character string indicating how the sampler is to be initialized. Only two options are currently supported: "prior" to initialize at a random draw from the prior; "RQ" to initialize at a model space approximation of the estimates from rq.
nknots	number of knots to be used for low rank approximation of the Gaussian process priors. Defaults to 6.
hyper	hyperparameters of the prior distribution. Must be a list with some of all of the following fields: sig: a two vector giving the parameters of the inverse-gamma distribution on sigma-square that is used when shrink=TRUE, lam: a two vector giving the parameters of the beta distribution on proximity = $\exp(-0.01 * \lambda^2)$, and kap: a vector to be coerced into a 3 * nkap matrix, with nkap being the number of components in the mixture of gamma prior on kappa, and each column of the matrix gives the shape, rate and mixing weight of a component.
shrink	for applying shrinkage to gamma[0] and gamma. Defaults to FALSE, in which case a right Haar prior is used on (gamma[0], gamma, sigma2). If TRUE then a horseshoe type prior is used.
prox.range	for specifying the range of length-scale parameter of the Gaussian process prior.
acpt.target	target acceptance rate of the adaptive Metropolis sampler; defaults to 0.15

`ref.size`	adaptation rate of the adaptive Metropolis sampler. The proposal density is updated once every `ref.size` iterations. Could be a single number or a vector of length same as the number of blocks.
`blocking`	type of blocking to be applied. Either a character string specifying one to be chosen from the supplied menu (see Details), or a list giving user specified blocks. In the latter case, each element of the list is a logical vector of length equal to the total number of model parameters, which equals `(nknots+1)*(ncol(X)+1) + 2` indicating which model parameters belong to the block.
`temp`	temperature of the log-likelihood function. The log-likelihood function is raised to the power of `temp`. Defaults to 1.
`expo`	the exponent to be used in the covariance kernel of the Gaussian process priors. Defaults to 2, giving the standard squared-exponential covariance kernel.
`blocks.mu`	initial block specific means in the form of a list. If left unspecified then will be automatically generated as a list of vectors of zeros of appropriate lengths matching the corresponding block sizes.
`blocks.S`	initial block specific covariance matrices in the form of a list. If left unspecified then will be automatically generated as a list of identity matrices of appropriate dimensions matching the corresponding block sizes. When `blocking` is chosen as one of the menu items of the form "std*", known prior covariance information and estimated variance matrices from `rq` are used.
`fix.nu`	either the logical FALSE indicating that nu should be learned, or a positive real number giving the fixed value of nu, which is then excluded from MCMC updates
`object`	a fitted model of the class 'qrjoint'.
`nadd`	number of additional MCMC samples.
`append`	logical indicating whether new samples should be appended to old ones. If FALSE then old samples are discarded.
`...`	no additional arguments are allowed

Details

The following menu choices are available for blocking the parameter vector. Below, `p = ncol(X)`.

`"single"`: a single block containing all parameters

`"single2"`: one block containing all parameters and an additional block containing only (gamma[0], gamma, sigma, nu)

`"single3"`: like `"single2"`, but the second block is split into two further blocks, one with (γ_0, γ), the other with (σ, ν)

`"std0"`: `p+1` blocks, $(j+1)$-th contains $(W_{*j}, \gamma_j, \sigma, \nu)$, $j = 0, \ldots, p$.

`"std1"`: total `p+2` blocks. First `p+1` blocks same as `"std0"` and one additional block of $(\gamma_0, \gamma, \sigma, \nu)$.

`"std2"`: total `p+3` blocks. First `p+1` blocks same as `"std0"` and two additional blocks of (γ_0, γ) and (σ, ν)

`"std3"`: total `p+3` blocks. First `p+1` blocks are W_{*j}, $j = 0, \ldots, p$, last two are (γ_0, γ) and (σ, ν)

`"std4"`: total `p+3` blocks. First `p+1` blocks are (W_{*j}, γ_j), $j = 0, \ldots, p$, last two are (γ_0, γ) and (σ, ν)

"std5": total p+4 blocks. First p+3 are same as "std4" and one additional block containing all parameters.

Value

qrjoint(x, y, ...) returns a 'qrjoint' class object to be used by update, coef and summary.

update(object,...) runs additional Markov chain iterations and appends thinned draws to an existing 'qrjoint' object object. All relevant details are inherited from object.

Returned object is a list containing the following variables.

par	latest draw of the parameter vector
x	scaled and centered design matrix
y	response vector
cens	censoring status vector
shrink	shrinkage indicator
hyper	completed list of hyper-parameters
dim	model dimension vector of the form c(n, p, length of tau grid, position of τ_0 on the grid, nknots, length of lambda grid, nkap, total number of MCMC iterations, thin, nsamp)
gridmats	details of covariance matrix factors etc, intended for internal use.
tau.g	the tau grid
muV	list of means for parameter blocks
SV	list of covariance matrices for parameter blocks
blocks	list of blocks
blocks.size	vector of block lengths
dmcmcpar	numeric vector containing details of adaptive MCMC runs, equals c(temp, decay rate of adaptation, vector of target acceptance rates for the blocks, vector of increment scales used in adaptation). Intended strictly for internal use.
imcmcpar	numeric vector containing details of adaptive MCMC runs, equals c(number of parameter blocks, ref.size, indicator on whether details are to be printed during MCMC progress, rate of details printing, a vector of counters needed for printing). Intended strictly for internal use.
parsamp	a long vector containing the parameter draws. Could be coerced into a matrix of dim npar * nsamp. Intended primarily for use by summary and coef.
acptsamp	a long vector containing rates of acceptance statistics for parameter blocks. Could be coerced into a matrix of dim nblocks * nsamp. Not very informative, because thinning times and adaptation times may not be exactly synced.
lpsamp	vector of log posterior values for the saved MCMC draws.
prox	vector of proximity (exp(-0.01*lambda^2)) grid values
reg.ix	positions of the regular tau grid on the expanded tail-appended grid
runtime	run time of the MCMC

References

Yang, Y., and Tokdar, S. T. (2015). Joint Estimation of Quantile Planes over Arbitrary Predictor Spaces.

See Also

summary.qrjoint and coef.qrjoint.

Examples

```
## Plasma data analysis

# recoding variables
data(plasma)
plasma$Sex <- as.factor(plasma$Sex)
plasma$SmokStat <- as.factor(plasma$SmokStat)
plasma$VitUse <- 3 - plasma$VitUse
plasma$VitUse <- as.factor(plasma$VitUse)

# creating predictors and response (beta carotene concentration in the plasma)
X <- model.matrix(BetaPlasma ~ Age + Sex + SmokStat + Quetelet + VitUse + Calories +
        Fat + Fiber + Alcohol + Cholesterol + BetaDiet, data = plasma)[,-1]
Y <- plasma$BetaPlasma

# model fitting with 40 posterior samples from 80 iterations (thin = 2)
# this is of course for illustration, for practical model fitting you
# ought to try at least something like nsamp = 500, thin = 20
fit.qrj <- qrjoint(X, Y, nsamp = 40, thin = 2)
summary(fit.qrj, more = TRUE)

## Not run:
# additional MCMC runs to get 10 more samples (20 additional iterations)
fit.qrj <- update(fit.qrj, 10)
summary(fit.qrj, more = TRUE)

## End(Not run)

## Not run:
### UIS data analysis (with right censoring)
data(uis)
y <- uis$Y
cens <- 1 - uis$CENSOR

X <- model.matrix(~ TREAT + NDT + IV3 + BECK + FRAC +
        RACE + AGE + SITE, data = uis)[,-1]
uis.qrj <- qrjoint(X, y, cens = cens, nsamp = 50, thin = 2, fix.nu = 1e5)
summary(uis.qrj, more = TRUE)

betas <- coef(uis.qrj, plot = TRUE, col = "darkgreen")
tau.grid <- uis.qrj$tau.g[uis.qrj$reg.ix]
L <- length(tau.grid)
```

```
beta.samp <- betas$beta.samp


# survival curve estimation for 9 randomly chosen subjects
n <- nrow(X)
ix.sel <- sort(sample(n, 9))
Xsel <- X[ix.sel,]
Qsel.gp <- apply(beta.samp, 2, function(b) return(tcrossprod(matrix(b, nrow = L), cbind(1, Xsel))))
k <- length(ix.sel)

colRGB <- col2rgb("darkgreen")/255
colTrans <- rgb(colRGB[1], colRGB[2], colRGB[3], alpha = 0.05)
par(mfrow = c(3,3), mar = c(4,3,2,1) + .1)
for(i in 1:k){
  plot(exp(Qsel.gp[(i-1)*L + 1:L,1]), 1 - tau.grid, ty = "n", ann = FALSE,
      bty = "n", xlim = exp(c(2, 8)), ylim = c(0,1), lty = 2, log = "x")
  for(j in 1:ncol(beta.samp))
      lines(exp(Qsel.gp[(i-1)*L + 1:L,j]), 1 - tau.grid, col = colTrans, lwd = 1)
  title(xlab = "Return time (days)", ylab = "Survival function", line = 2)
  title(main = bquote(Obs.Id == .(ix.sel[i])))
  grid()
}

## End(Not run)
```

---

summary.qrjoint *Summary Method for qrjoint Model Fit*

Description

Summarize model fit, including MCMC details, for qrjoint

Usage

```
## S3 method for class 'qrjoint'
summary(object, ntrace = 1000, plot.dev = TRUE, more.details = FALSE, ...)
```

Arguments

object	a fitted model of the class 'qrjoint'.
ntrace	number of draws to be included in trace plots
plot.dev	logical indicator of whether to show trace plot of deviance
more.details	logical indicating whether other details from MCMC are to be plotted
...	a limited number of plotting controls that are passed onto the deviance plot

Value

Displays the trace of the deviance statistic. More details include trace plots of of the proximity parameter of each GP, a plot of Geweke p-values for (from `geweke.diag`) convergence of each model parameter and an image plot of parameter correlation. Also prints two versions of Watanabe AIC.

The following quantities are returned invisibly.

`deviance`	vector deviance statistic of the samples parameter draws
`pg`	a matrix with `nsamp` number of columns, each columns could be coerced into a matrix of dimension `ngrid * (p+1)`, where the columns gives the conditional posterior weights on the lambda grid values for the corresponding GP function.
`prox`	posterior draws of proximity in the form of a `(p+1)*nsamp` matrix.
`ll`	a matrix of `n*nsamp` containing observation level log-likelihood contributions. Used to calculate *waic*, and could be used for other AIC calculations.
`waic`	Two versions of Watanabe AIC from Gelman, Hwang and Vehtari (2014).

References

Gelman, A., Hwang, J., and Vehtari, A. (2014). Understanding predictive information criterion for Bayesian models. *Stat Comput*, 24, 997-1016.

See Also

`qrjoint` and `coef.qrjoint`.

Examples

```
# Plasma data analysis

# recoding variables
data(plasma)
plasma$Sex <- as.factor(plasma$Sex)
plasma$SmokStat <- as.factor(plasma$SmokStat)
plasma$VitUse <- 3 - plasma$VitUse
plasma$VitUse <- as.factor(plasma$VitUse)

# creating predictors and response (beta carotene concentration in the plasma)
X <- model.matrix(BetaPlasma ~ Age + Sex + SmokStat + Quetelet + VitUse + Calories +
        Fat + Fiber + Alcohol + Cholesterol + BetaDiet, data = plasma)[,-1]
Y <- plasma$BetaPlasma

# model fitting with 50 posterior samples from 100 iterations (thin = 2)
fit.qrj <- qrjoint(X, Y, 50, 2)
summary(fit.qrj, more = TRUE)
```

waic *Watanabe Information Criterion*

Description

Calculates two versions of the Watanabe information criteria from MCMC draws.

Usage

```
waic(logliks, print = TRUE)
```

Arguments

logliks	a matrix of observation level log-likelihood values, the columns are MCMC iterations and the rows are observations in the data
print	logical whether to print the results

Value

Returns the two version of the WAIC

References

Gelman, A., Hwang, J., and Vehtari, A. (2014). Understanding predictive information criterion for Bayesian models. *Stat Comput*, 24, 997-1016.

See Also

summary.qrjoint

Examples

```
# Plasma data analysis

# recoding variables
data(plasma)
plasma$Sex <- as.factor(plasma$Sex)
plasma$SmokStat <- as.factor(plasma$SmokStat)
plasma$VitUse <- 3 - plasma$VitUse
plasma$VitUse <- as.factor(plasma$VitUse)

# creating predictors and response (beta carotene concentration in the plasma)
X <- model.matrix(BetaPlasma ~ Age + Sex + SmokStat + Quetelet + VitUse + Calories +
        Fat + Fiber + Alcohol + Cholesterol + BetaDiet, data = plasma)[,-1]
Y <- plasma$BetaPlasma

# model fitting with 50 posterior samples from 100 iterations (thin = 2)
```

```
fit.qrj <- qrjoint(X, Y, 50, 2)
sm <- summary(fit.qrj, plot = FALSE)

# the call to summary already shows the waic for the fitted model, it also returns
# the observation level log-likelihood vales. To calculate waic from last 20 draws
# we can use:

ll <- sm$ll
waic(ll[,31:50])
```

Index