Package ‘DRclass’

March 10, 2025

Type Package

Title Consider Ambiguity in Probabilistic Descriptions Using Density Ratio Classes

Version 0.1.0

Description Consider ambiguity in probabilistic descriptions by replacing a parametric probabilistic description of uncertainty by a non-parametric set of probability distributions in the form of a Density Ratio Class. This is of particular interest in Bayesian inference. The Density Ratio Class is particularly suited for this purpose as it is invariant under Bayesian inference, marginalization, and propagation through a deterministic model. Here, invariant means that the result of the operation applied to a Density Ratio Class is again a Density Ratio Class. In particular the invariance under Bayesian inference thus enables iterative learning within the same framework of Density Ratio Classes.
The use of imprecise probabilities in general, and Density Ratio Classes in particular, lead to intervals of characteristics of probability distributions, such as cumulative distribution functions, quantiles, and means.
The package is based on a sample of the distribution proportional to the upper bound of the class. Typically this will be a sample from the posterior in Bayesian inference. Based on such a sample, the package provides functions to calculate lower and upper class boundaries and lower and upper bounds of cumulative distribution functions, and quantiles.
Rinderknecht, S.L., Albert, C., Borsuk, M.E., Schuwirth, N., Kuensch, H.R. and Reichert, P. (2014) ``The effect of ambiguous prior knowledge on Bayesian model parameter inference and prediction." Environmental Modelling & Software. 62, 300-315, 2014. <doi:10.1016/j.envsoft.2014.08.020>.
Sriwastava, A. and Reichert, P. ``Robust Bayesian Estimation of Value Function Parameters using Imprecise Priors." Submitted. <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4973574>.

License GPL-3

Depends stats

URL https://gitlab.com/p.reichert/DRclass

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Peter Reichert [aut, cre],
Ambuj Sriwastava [aut]

Maintainer Peter Reichert <peter.reichert@emeriti.eawag.ch>

Repository CRAN

Date/Publication 2025-03-10 15:20:07 UTC

Contents

`DRclass-package` *Consider Ambiguity in Probabilistic Descriptions Using Density Ratio Classes*

Description

Consider ambiguity in probabilistic descriptions by replacing a parametric probabilistic description of uncertainty by a non-parametric set of probability distributions in the form of a Density Ratio Class. This is of particular interest in Bayesian inference.

The Density Ratio Class is particularly suited for this purpose as it is invariant under Bayesian inference, marginalization, and propagation through a deterministic model. Here, invariant means that the result of the operation applied to a Density Ratio Class is again a Density Ratio Class. In particular the invariance under Bayesian inference thus enables iterative learning within the same framework of Density Ratio Classes.

The replacement of exact priors by Density Ratio Classes makes it possible to determine the sensitivity of the posterior to the shape of the prior. This is an important extension of the investigation to the width of the prior.

The use of imprecise probabilities in general, and Density Ratio Classes in particular, lead to intervals of characteristics of probability distributions, such as cumulative distribution functions, quantiles, and means.

The package is based on a sample of distribution proportional to the upper bound of the class. Typically this will be sample from the posterior in Bayesian inference. Based on such a sample, the package provides functions to calculate
- lower and upper class boundaries (`DRclass_k_Pdf`, `DRclass_lu_Pdf`),
- lower and upper bounds of cumulative distribution functions (`DRclass_k_Cdf`, `DRclass_lu_Cdf`),
- lower and upper bounds of quantiles (`DRclass_k_Quantile`, `DRclass_lu_Quantile`), and

for the marginals of the Density Ratio Class.

This sensitivity analysis to the shape of the distribution is obtained computationally very cheaply even for the results of Bayesian inference. If the bounding functions of the prior class are proportional, this does not require any additional evaluation of the bounding functions, l and u, once a sample of the posterior of the distribution proportional to the upper bounding function is available (functions `DRclass_k_Pdf`, `DRclass_k_Cdf`, `DRclass_k_Quantile`). In the case of different shapes, the bounding functions have to be evaluated at the sample points (functions `DRclass_lu_Pdf`, `DRclass_lu_Cdf`, `DRclass_lu_Quantile`). However, as only the ratio of bounding functions, l and u, is needed, in the context of Bayesian inference, the bounding functions of the prior can be provided as the likelihood, L, cancels when calculating the ratio: (l L)/(u L) = l/u. As in most cases the evaluation of the prior is computationally much cheaper than that of the likelihood, the computational effort for the sensitivity analysis is much smaller than the effort of the preceding sampling from the posterior in Bayesian inference..

The package contains a copy of the bisection algorithm from package cmna (https://cran.r-project.org/package=cmna) modified to allow for additional arguments to the function of which the root is to be found.

References

Direct references underlying this package:

Rinderknecht, S.L., Albert, C., Borsuk, M.E., Schuwirth, N., Kuensch, H.R. and Reichert, P. The effect of ambiguous prior knowledge on Bayesian model parameter inference and prediction. Environmental Modelling & Software. 62, 300-315, 2014. https://doi.org/10.1016/j.envsoft.2014.08.020.

Sriwastava, A. and Reichert, P. Robust Bayesian Estimation of Value Function Parameters using Imprecise Priors. Submitted. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4973574.

Original references to the Density Ratio Class:

DeRobertis, L., Hartigan, J.A. Bayesian inference using intervals of measures. Ann. Stat. 9 (2), 235-244, 1981.

Berger, J.O. Robust Bayesian analysis: sensitivity to the prior. J. Stat. Plan. Inference 25, 303-328, 1990.

Wasserman, L. Invariance properties of density ratio priors. Ann. Stat. 20 (4), 2177-2182, 1992.

Wasserman, L. and Kadane, J. B. Computing bounds of expectations. Journal of the Americal Statistical Association, 87(418), 483–503, 1992.

Berger, J.O. An overview of robust Bayesian analysis. Test 3, 5-124, 1994.

Geweke, J. and Petrella, L. Prior density-ratio class robustness in econometrics. Journal of Business & Economic Statistics, 16(4), 469–478, 1998.

`DRclass_k_Cdf` *Calculate lower and upper bounds of the cumulative density function for the marginals of a Density Ratio Class for which the lower and upper bounding functions are proportional.*

Description

The cumulative density functions for the marginals of a distributions proportional to the lower and upper bounding functions are also provided.

Usage

```
DRclass_k_Cdf(sample_u, k = 1, nout = 512)
```

Arguments

sample_u	Sample from a distribution proportional to the upper bound of the class, often from the posterior of the upper bound of the prior in Bayesian inference. Columns represent variables, rows go across the sample.
k	Factor of proportionality between upper (u) and lower (l) bound: u = k * l
nout	Number of equally spaced output intervals for the marginal densities.

Details

This function is more efficient than 'DRclass_lu_Pdf' as it does not need the evaluation of the bounding functions, l and u. It is thus recommended to use this function if l and u are proportional.

Value

Three dimensional array with the following dimensions: 1: variable corresponding to column of the sample 2: equidistant spacing of that variable 3: five columns for variable values, lower bound of the cdf of the marginal of the class, cdf of the marginal corresponding to the lower class boundary, cdf of the marginal corresponding to the upper class boundary, upper bound of the cdf of the marginal of the class

Examples

```
# example of the application of DRclass functions:
# ------------------------------------------------

# parameter values
k        <- 10
sd       <- 0.5
sampsize <- 10000

# upper and lower class boundaries:
u <- function(x) { return(    dnorm(x,0,sd)) }
l <- function(x) { return(1/k*dnorm(x,0,sd)) }

# generate sample:
sample_u <- cbind(rnorm(sampsize,0,sd),rnorm(sampsize,0,sd))  # example of 2d sample

# get class boundaries (back from sample):
pdf1   <- DRclass_k_Pdf(sample_u,k=k,adjust=2)       # faster for l proportional to u
pdf2  <- DRclass_lu_Pdf(sample_u,l=l,u=u,adjust=2)  # l and u could have different shapes
```

```
# get cdf bounds:
cdf1   <- DRclass_k_Cdf(sample_u,k=k)
cdf2   <- DRclass_lu_Cdf(sample_u,l=l,u=u)

# get quantile bounds:
quant1 <- DRclass_k_Quantile(sample_u,k=k,probs=c(0.025,0.5,0.975))
quant2 <- DRclass_lu_Quantile(sample_u,l=l,u=u,probs=c(0.025,0.5,0.975))

# plot selected features of the first component of the sample:
oldpar <- par(no.readonly=TRUE)
par(mar=c(5, 4, 1, 4) + 0.1)  # c(bottom, left, top, right)
plot(pdf1[1,,c("x","u")],type="l",xaxs="i",yaxs="i",xlim=c(-2,2),xlab="x",ylab="pdf")
lines(pdf2[1,,c("x","l")])
par(new=TRUE)
plot(cdf1[1,,c("x","F_upper")],type="l",xaxs="i",yaxs="i",axes=FALSE,
     xlim=c(-2,2),ylim=c(0,1),ylab="",lty="dashed")
axis(4); mtext("cdf",4,2)
lines(cdf2[1,,c("x","F_lower")],lty="dashed")
abline(v=quant1["quant_lower_0.5",1],lty="dotted")
abline(v=quant1["quant_upper_0.5",1],lty="dotted")
abline(v=quant1["quant_lower_0.025",1],lty="dotdash")
abline(v=quant1["quant_upper_0.975",1],lty="dotdash")
par(oldpar)
```

DRclass_k_Pdf *Calculate marginal class bounding functions for a special Density Ratio Class for which the lower and upper bounding functions are proportional.*

Description

This function is more efficient than 'DRclass_lu_Pdf' as it does not need the evaluation of the bounding functions, l and u. It is thus recommended to use this function if l and u are proportional.

Usage

```
DRclass_k_Pdf(sample_u, k = 1, nout = 512, ...)
```

Arguments

sample_u	Sample from a distribution proportional to the upper bound of the class, often from the posterior of the upper bound of the prior in Bayesian inference. Columns represent variables, rows go across the sample.
k	Factor of proportionality between upper (u) and lower (l) bound: u = k * l
nout	Number of equally spaced output intervals for the marginal densities.
...	Further arguments passed to the function 'density'

Value

Three dimensional array with the following dimensions: 1: variable corresponding to column of the sample 2: equidistant spacing of that variable 3: three columns for variable values, upper normalized density of the marginal class, lower non-normalized density of the marginal class

Examples

```
# example of the application of DRclass functions:
# -----------------------------------------------

# parameter values
k        <- 10
sd       <- 0.5
sampsize <- 10000

# upper and lower class boundaries:
u <- function(x) { return(    dnorm(x,0,sd)) }
l <- function(x) { return(1/k*dnorm(x,0,sd)) }

# generate sample:
sample_u <- cbind(rnorm(sampsize,0,sd),rnorm(sampsize,0,sd))  # example of 2d sample

# get class boundaries (back from sample):
pdf1   <- DRclass_k_Pdf(sample_u,k=k,adjust=2)       # faster for l proportional to u
pdf2   <- DRclass_lu_Pdf(sample_u,l=l,u=u,adjust=2)  # l and u could have different shapes

# get cdf bounds:
cdf1   <- DRclass_k_Cdf(sample_u,k=k)
cdf2   <- DRclass_lu_Cdf(sample_u,l=l,u=u)

# get quantile bounds:
quant1 <- DRclass_k_Quantile(sample_u,k=k,probs=c(0.025,0.5,0.975))
quant2 <- DRclass_lu_Quantile(sample_u,l=l,u=u,probs=c(0.025,0.5,0.975))

# plot selected features of the first component of the sample:
oldpar <- par(no.readonly=TRUE)
par(mar=c(5, 4, 1, 4) + 0.1)  # c(bottom, left, top, right)
plot(pdf1[1,,c("x","u")],type="l",xaxs="i",yaxs="i",xlim=c(-2,2),xlab="x",ylab="pdf")
lines(pdf2[1,,c("x","l")])
par(new=TRUE)
plot(cdf1[1,,c("x","F_upper")],type="l",xaxs="i",yaxs="i",axes=FALSE,
     xlim=c(-2,2),ylim=c(0,1),ylab="",lty="dashed")
axis(4); mtext("cdf",4,2)
lines(cdf2[1,,c("x","F_lower")],lty="dashed")
abline(v=quant1["quant_lower_0.5",1],lty="dotted")
abline(v=quant1["quant_upper_0.5",1],lty="dotted")
abline(v=quant1["quant_lower_0.025",1],lty="dotdash")
abline(v=quant1["quant_upper_0.975",1],lty="dotdash")
par(oldpar)
```

DRclass_k_Quantile *Calculate lower and upper bounds of quantiles of marginals of a Density Ratio Class for which the lower and upper bounding functions are proportional.*

Description

Quantiles of the marginals of distributions proportional to the lower and upper bounding functions are also provided.

Usage

```
DRclass_k_Quantile(
  sample_u,
  k = 1,
  probs = c(0.025, 0.25, 0.5, 0.75, 0.975),
  tol = 0.001,
  maxiter = 100
)
```

Arguments

`sample_u` Sample from a distribution proportional to the upper bound of the class, often from the posterior of the upper bound of the prior in Bayesian inference. Columns represent variables, rows go across the sample.

`k` Factor of proportionality between upper (u) and lower (l) bound: u = k * l

`probs` Vector of probabilities for which the quantile bounds are to be provided.

`tol` Tolerance in quantile value for approximating the solution of the implicit equation for quantiles with the bisection algorithm.

`maxiter` Maximum number of iterations for approximating the solution of the implicit equation for quantiles with the bisection algorithm.

Details

This function is more efficient than ’DRclass_lu_Pdf’ as it does not need the evaluation of the bounding functions, l and u. It is thus recommended to use this function if l and u are proportional.

Value

Matrix of quantile bounds and quantiles (rows) for different marginal variables (columns).

Examples

```
# example of the application of DRclass functions:
# ------------------------------------------------

# parameter values
```

```
k        <- 10
sd       <- 0.5
sampsize <- 10000

# upper and lower class boundaries:
u <- function(x) { return(    dnorm(x,0,sd)) }
l <- function(x) { return(1/k*dnorm(x,0,sd)) }

# generate sample:
sample_u <- cbind(rnorm(sampsize,0,sd),rnorm(sampsize,0,sd))  # example of 2d sample

# get class boundaries (back from sample):
pdf1   <- DRclass_k_Pdf(sample_u,k=k,adjust=2)       # faster for l proportional to u
pdf2  <- DRclass_lu_Pdf(sample_u,l=l,u=u,adjust=2)  # l and u could have different shapes

# get cdf bounds:
cdf1   <- DRclass_k_Cdf(sample_u,k=k)
cdf2   <- DRclass_lu_Cdf(sample_u,l=l,u=u)

# get quantile bounds:
quant1 <- DRclass_k_Quantile(sample_u,k=k,probs=c(0.025,0.5,0.975))
quant2 <- DRclass_lu_Quantile(sample_u,l=l,u=u,probs=c(0.025,0.5,0.975))

# plot selected features of the first component of the sample:
oldpar <- par(no.readonly=TRUE)
par(mar=c(5, 4, 1, 4) + 0.1)  # c(bottom, left, top, right)
plot(pdf1[1,,c("x","u")],type="l",xaxs="i",yaxs="i",xlim=c(-2,2),xlab="x",ylab="pdf")
lines(pdf2[1,,c("x","l")])
par(new=TRUE)
plot(cdf1[1,,c("x","F_upper")],type="l",xaxs="i",yaxs="i",axes=FALSE,
     xlim=c(-2,2),ylim=c(0,1),ylab="",lty="dashed")
axis(4); mtext("cdf",4,2)
lines(cdf2[1,,c("x","F_lower")],lty="dashed")
abline(v=quant1["quant_lower_0.5",1],lty="dotted")
abline(v=quant1["quant_upper_0.5",1],lty="dotted")
abline(v=quant1["quant_lower_0.025",1],lty="dotdash")
abline(v=quant1["quant_upper_0.975",1],lty="dotdash")
par(oldpar)
```

DRclass_lu_Cdf

Calculate lower and upper bounds of the cumulative density function for the marginals of a general Density Ratio Class. The cumulative density functions for the marginals of a distributions proportional to the lower and upper bounding functions are also provided. See the function 'DRclass_k_Cdf' for the case in which the upper and lower bounding functions of the class are proportional.

Description

Calculate lower and upper bounds of the cumulative density function for the marginals of a general Density Ratio Class. The cumulative density functions for the marginals of a distributions proportional to the lower and upper bounding functions are also provided. See the function 'DRclass_k_Cdf' for the case in which the upper and lower bounding functions of the class are proportional.

Usage

```
DRclass_lu_Cdf(sample_u, l, u, nout = 512)
```

Arguments

`sample_u` Sample from a distribution proportional to the upper bound of the class, often from the posterior of the upper bound of the prior in Bayesian inference. Columns represent variables, rows go across the sample.

`l` Either a function to evaluate the lower bound of the Density Ratio Class or a vector of values of this function evaluated for the rows of 'sample_u'. Note that in the context of Bayesian inference the upper bound of the prior can be provided as only the ratio of l/u is needed and the likelihood cancels in this fraction. This saves computation time as the prior is usually computationally much cheaper to evaluate than the likelihood.

`u` Either a function to evaluate the upper bound of the Density Ratio Class or a vector of values of this function evaluated for the rows of 'sample_u'. Note that in the context of Bayesian inference the lower bound of the prior can be provided as only the ratio of l/u is needed and the likelihood cancels in this fraction. This saves computation time as the prior is usually computationally much cheaper to evaluate than the likelihood.

`nout` Number of equally spaced output intervals for the marginal densities.

Value

Three dimensional array with the following dimensions: 1: variable corresponding to column of the sample 2: equidistant spacing of that variable 3: five columns for variable values, lower bound of the cdf of the marginal of the class, cdf of the marginal corresponding to the lower class boundary, cdf of the marginal corresponding to the upper class boundary, upper bound of the cdf of the marginal of the class

Examples

```
# example of the application of DRclass functions:
# ----------------------------------------------

# parameter values
k        <- 10
sd       <- 0.5
sampsize <- 10000

# upper and lower class boundaries:
u <- function(x) { return(    dnorm(x,0,sd)) }
l <- function(x) { return(1/k*dnorm(x,0,sd)) }

# generate sample:
```

```
sample_u <- cbind(rnorm(sampsize,0,sd),rnorm(sampsize,0,sd))  # example of 2d sample

# get class boundaries (back from sample):
pdf1   <- DRclass_k_Pdf(sample_u,k=k,adjust=2)       # faster for l proportional to u
pdf2   <- DRclass_lu_Pdf(sample_u,l=l,u=u,adjust=2)  # l and u could have different shapes

# get cdf bounds:
cdf1   <- DRclass_k_Cdf(sample_u,k=k)
cdf2   <- DRclass_lu_Cdf(sample_u,l=l,u=u)

# get quantile bounds:
quant1 <- DRclass_k_Quantile(sample_u,k=k,probs=c(0.025,0.5,0.975))
quant2 <- DRclass_lu_Quantile(sample_u,l=l,u=u,probs=c(0.025,0.5,0.975))

# plot selected features of the first component of the sample:
oldpar <- par(no.readonly=TRUE)
par(mar=c(5, 4, 1, 4) + 0.1)  # c(bottom, left, top, right)
plot(pdf1[1,,c("x","u")],type="l",xaxs="i",yaxs="i",xlim=c(-2,2),xlab="x",ylab="pdf")
lines(pdf2[1,,c("x","l")])
par(new=TRUE)
plot(cdf1[1,,c("x","F_upper")],type="l",xaxs="i",yaxs="i",axes=FALSE,
     xlim=c(-2,2),ylim=c(0,1),ylab="",lty="dashed")
axis(4); mtext("cdf",4,2)
lines(cdf2[1,,c("x","F_lower")],lty="dashed")
abline(v=quant1["quant_lower_0.5",1],lty="dotted")
abline(v=quant1["quant_upper_0.5",1],lty="dotted")
abline(v=quant1["quant_lower_0.025",1],lty="dotdash")
abline(v=quant1["quant_upper_0.975",1],lty="dotdash")
par(oldpar)
```

DRclass_lu_Pdf *Calculate marginal class bounding functions for the general case of a Density Ratio Class.*

Description

See the function 'DRclass_k_Pdf' for the case in which the upper and lower bounding functions of the class are proportional.

Usage

```
DRclass_lu_Pdf(sample_u, l, u, nout = 512, ...)
```

Arguments

sample_u	Sample from a distribution proportional to the upper bound of the class, often from the posterior of the upper bound of the prior in Bayesian inference. Columns represent variables, rows go across the sample.

`l` Either a function to evaluate the lower bound of the Density Ratio Class or a vector of values of this function evaluated for the rows of 'sample_u'. Note that in the context of Bayesian inference the upper bound of the prior can be provided as only the ratio of l/u is needed and the likelihood cancels in this fraction. This saves computation time as the prior is usually computationally much cheaper to evaluate than the likelihood.

`u` Either a function to evaluate the upper bound of the Density Ratio Class or a vector of values of this function evaluated for the rows of 'sample_u'. Note that in the context of Bayesian inference the lower bound of the prior can be provided as only the ratio of l/u is needed and the likelihood cancels in this fraction. This saves computation time as the prior is usually computationally much cheaper to evaluate than the likelihood.

`nout` Number of equally spaced output intervals for the marginal densities.

`...` Further arguments passed to the function 'density'

Value

Three dimensional array with the following dimensions: 1: variable corresponding to column of the sample 2: equidistant spacing of that variable 3: three columns for variable values, upper normalized density of the marginal class, lower non-normalized density of the marginal class

Examples

```
# example of the application of DRclass functions:
# -----------------------------------------------

# parameter values
k        <- 10
sd       <- 0.5
sampsize <- 10000

# upper and lower class boundaries:
u <- function(x) { return(    dnorm(x,0,sd)) }
l <- function(x) { return(1/k*dnorm(x,0,sd)) }

# generate sample:
sample_u <- cbind(rnorm(sampsize,0,sd),rnorm(sampsize,0,sd))  # example of 2d sample

# get class boundaries (back from sample):
pdf1   <- DRclass_k_Pdf(sample_u,k=k,adjust=2)       # faster for l proportional to u
pdf2  <- DRclass_lu_Pdf(sample_u,l=l,u=u,adjust=2)  # l and u could have different shapes

# get cdf bounds:
cdf1   <- DRclass_k_Cdf(sample_u,k=k)
cdf2   <- DRclass_lu_Cdf(sample_u,l=l,u=u)

# get quantile bounds:
quant1 <- DRclass_k_Quantile(sample_u,k=k,probs=c(0.025,0.5,0.975))
quant2 <- DRclass_lu_Quantile(sample_u,l=l,u=u,probs=c(0.025,0.5,0.975))
```

```
# plot selected features of the first component of the sample:
oldpar <- par(no.readonly=TRUE)
par(mar=c(5, 4, 1, 4) + 0.1)  # c(bottom, left, top, right)
plot(pdf1[1,,c("x","u")],type="l",xaxs="i",yaxs="i",xlim=c(-2,2),xlab="x",ylab="pdf")
lines(pdf2[1,,c("x","l")])
par(new=TRUE)
plot(cdf1[1,,c("x","F_upper")],type="l",xaxs="i",yaxs="i",axes=FALSE,
     xlim=c(-2,2),ylim=c(0,1),ylab="",lty="dashed")
axis(4); mtext("cdf",4,2)
lines(cdf2[1,,c("x","F_lower")],lty="dashed")
abline(v=quant1["quant_lower_0.5",1],lty="dotted")
abline(v=quant1["quant_upper_0.5",1],lty="dotted")
abline(v=quant1["quant_lower_0.025",1],lty="dotdash")
abline(v=quant1["quant_upper_0.975",1],lty="dotdash")
par(oldpar)
```

DRclass_lu_Quantile

Calculate lower and upper bounds of quantiles of marginals of a Density Ratio Class. Quantiles of the marginals of distributions proportional to the lower and upper bounding functions are also provided. See the function 'DRclass_k_Quantile' for the case in which the upper and lower bounding functions of the class are proportional.

Description

Calculate lower and upper bounds of quantiles of marginals of a Density Ratio Class. Quantiles of the marginals of distributions proportional to the lower and upper bounding functions are also provided. See the function 'DRclass_k_Quantile' for the case in which the upper and lower bounding functions of the class are proportional.

Usage

```
DRclass_lu_Quantile(
  sample_u,
  l,
  u,
  probs = c(0.025, 0.25, 0.5, 0.75, 0.975),
  tol = 0.001,
  maxiter = 100
)
```

Arguments

`sample_u` Sample from a distribution proportional to the upper bound of the class, often from the posterior of the upper bound of the prior in Bayesian inference. Columns represent variables, rows go across the sample.

`l` Either a function to evaluate the lower bound of the Density Ratio Class or a vector of values of this function evaluated for the rows of 'sample_u'. Note that in the context of Bayesian inference the upper bound of the prior can be provided as only the ratio of l/u is needed and the likelihood cancels in this fraction. This saves computation time as the prior is usually computationally much cheaper to evaluate than the likelihood.

`u` Either a function to evaluate the upper bound of the Density Ratio Class or a vector of values of this function evaluated for the rows of 'sample_u'. Note that in the context of Bayesian inference the lower bound of the prior can be provided as only the ratio of l/u is needed and the likelihood cancels in this fraction. This saves computation time as the prior is usually computationally much cheaper to evaluate than the likelihood.

`probs` Vector of probabilities for which the quantile bounds are to be provided.

`tol` Tolerance in quantile value for approximating the solution of the implicit equation for quantiles with the bisection algorithm.

`maxiter` Maximum number of iterations for approximating the solution of the implicit equation for quantiles with the bisection algorithm.

Value

Matrix of quantile bounds and quantiles (rows) for different marginal variables (columns).

Examples

```
# example of the application of DRclass functions:
# -----------------------------------------------

# parameter values
k        <- 10
sd       <- 0.5
sampsize <- 10000

# upper and lower class boundaries:
u <- function(x) { return(    dnorm(x,0,sd)) }
l <- function(x) { return(1/k*dnorm(x,0,sd)) }

# generate sample:
sample_u <- cbind(rnorm(sampsize,0,sd),rnorm(sampsize,0,sd))  # example of 2d sample

# get class boundaries (back from sample):
pdf1   <- DRclass_k_Pdf(sample_u,k=k,adjust=2)       # faster for l proportional to u
pdf2  <- DRclass_lu_Pdf(sample_u,l=l,u=u,adjust=2)  # l and u could have different shapes

# get cdf bounds:
cdf1   <- DRclass_k_Cdf(sample_u,k=k)
cdf2   <- DRclass_lu_Cdf(sample_u,l=l,u=u)

# get quantile bounds:
quant1 <- DRclass_k_Quantile(sample_u,k=k,probs=c(0.025,0.5,0.975))
quant2 <- DRclass_lu_Quantile(sample_u,l=l,u=u,probs=c(0.025,0.5,0.975))
```

```
# plot selected features of the first component of the sample:
oldpar <- par(no.readonly=TRUE)
par(mar=c(5, 4, 1, 4) + 0.1)  # c(bottom, left, top, right)
plot(pdf1[1,,c("x","u")],type="l",xaxs="i",yaxs="i",xlim=c(-2,2),xlab="x",ylab="pdf")
lines(pdf2[1,,c("x","l")])
par(new=TRUE)
plot(cdf1[1,,c("x","F_upper")],type="l",xaxs="i",yaxs="i",axes=FALSE,
     xlim=c(-2,2),ylim=c(0,1),ylab="",lty="dashed")
axis(4); mtext("cdf",4,2)
lines(cdf2[1,,c("x","F_lower")],lty="dashed")
abline(v=quant1["quant_lower_0.5",1],lty="dotted")
abline(v=quant1["quant_upper_0.5",1],lty="dotted")
abline(v=quant1["quant_lower_0.025",1],lty="dotdash")
abline(v=quant1["quant_upper_0.975",1],lty="dotdash")
par(oldpar)
```

Index