Package ‘dsfa’

September 7, 2022

Title Distributional Stochastic Frontier Analysis

Version 0.0.3

Description Framework to fit distributional stochastic frontier models. Casts the stochastic frontier model into the flexible framework of distributional regression or otherwise known as General Additive Models of Location, Scale and Shape (GAMLSS). Allows for linear, non-linear, random and spatial effects on all the parameters of the distribution of the output, e.g. effects on the production or cost function, heterogeneity of the noise and inefficiency. Available distributions are the normal-halfnormal and normal-exponential distribution. Estimation via the fast and reliable routines of the 'mgcv' package. For more details see Schmidt R, Kneib T (2022) <doi:10.48550/arXiv.2208.10294>.

Imports mgcv, sn, gratia, graphics, stats, copula, numDeriv, Rdpack

NeedsCompilation yes

SystemRequirements C++11

License MIT + file LICENSE

Encoding UTF-8

RdMacros Rdpack

RoxygenNote 7.2.1

Author Rouven Schmidt [aut, cre]

Maintainer Rouven Schmidt <rouven.schmidt@tu-clausthal.de>

Repository CRAN

Date/Publication 2022-09-07 08:30:08 UTC

R topics documented:

`chainrule` *Chainrule*

Description

Calculates the partial derivatives of the function $h(x_1, x_2, ..., x_K) = f(g(x_1, x_2, ..., x_K))$ up to order four. Here K is the number of inputs for function $g(\cdot)$. The function $f(\cdot)$ can only have a single input.

Usage

```
chainrule(
  f1 = NULL,
  f2 = NULL,
  f3 = NULL,
  f4 = NULL,
  g1 = NULL,
  g2 = NULL,
  g3 = NULL,
  g4 = NULL,
  deriv = 2,
  xg = NULL
)
```

Arguments

`f1`	vector of first derivatives of $f(\cdot)$ evaluated at $g(\cdot)$.
`f2`	vector of second derivatives of $f(\cdot)$ evaluated at $g(\cdot)$.
`f3`	vector of third derivatives of $f(\cdot)$ evaluated at $g(\cdot)$.
`f4`	vector of fourth derivatives of $f(\cdot)$ evaluated at $g(\cdot)$.
`g1`	matrix of first derivatives of $g(\cdot)$.
`g2`	matrix of second derivatives of $g(\cdot)$.
`g3`	matrix of third derivatives of $g(\cdot)$.
`g4`	matrix of fourth derivatives of $g(\cdot)$.
`deriv`	derivative of order `deriv`. Available are 1,2,3,4.
`xg`	optional, index arrays for upper triangular for g, generated by `trind.generator`.

Details

Mostly internal function, which is helpful in calculating the partial derivatives of the loglikelihood.

Value

A list with partial derivatives. The index of the list corresponds to a matrix with all partial derivatives of that order.

Examples

```
x<-1 #For K=1, x_1 value is set to 1.

g<-1/x #g(x_1) = 1/x
g1<-matrix(-1/x^2,ncol=1)
g2<-matrix(2/x^3,ncol=1)
g3<-matrix(-6/x^4,ncol=1)
g4<-matrix(24/x^5,ncol=1)

f<-exp(g) #f(g(x)) = exp(g(x))
f1<-f2<-f3<-f4<-exp(g)

chainrule(f1, f2, f3, f4, g1, g2, g3, g4, deriv=4)
```

---

comperr_mv — *Composed error multivariate distribution object for mgcv*

Description

The comperr_mv family implements the composed error multivariate distribution in which the μ_1, σ_{V1}, σ_{U1}, (or λ_1), μ_2, σ_{V2}, σ_{U2}, (or λ_2) and τ can depend on additive predictors. Useable only with `mgcv::gam`, the additive predictors are specified via a list of formulae.

Usage

```
comperr_mv(
  link = list("identity", "log", "log", "identity", "log", "log", "tanh"),
  s1 = -1,
  s2 = -1,
  dist = c("normhnorm", "normhnorm", 1)
)
```

Arguments

`link`	seven item list specifying the link for the μ_1, σ_{V1}, σ_{U1}, λ_1, μ_2, σ_{V2}, σ_{U2}, λ_2 and τ parameters. See details.
`s1`	$s_1 = -1$ for production and $s_1 = 1$ for cost function for margin 1.
`s2`	$s_2 = -1$ for production and $s_2 = 1$ for cost function for margin 2.

`dist` vector of length three, specifying the bivariate distribution. First element is the name of the first marginal distribution. Second element is the name of the second marginal distribution. Third element specifies the copula. See dcop for more details.

Details

Used with gam to fit distributional stochastic frontier model. The function `gam` is from the mgcv package is called with a list containing seven formulae:

1. The first formula specifies the response of margin 1 on the left hand side and the structure of the additive predictor for μ_1 parameter on the right hand side. Link function is "identity".
2. The second formula is one sided, specifying the additive predictor for the σ_{V1} on the right hand side. Link function is "log".
3. The third formula is one sided, specifying the additive predictor for the σ_{U1} or λ_1 on the right hand side. Link function is "log".
4. The fourth formula specifies the response of margin 2 on the left hand side and the structure of the additive predictor for μ_2 parameter on the right hand side. Link function is "identity".
5. The fifth formula is one sided, specifying the additive predictor for the σ_{V2} on the right hand side. Link function is "log".
6. The sixth formula is one sided, specifying the additive predictor for the σ_{U2} or λ_2 on the right hand side. Link function is "log".
7. The seventh formula is one sided, specifying the additive predictor for the τ on the right hand side. Link function is "tanh".

The fitted values and linear predictors for this family will be seven column matrices. The first column is the μ_1, the second column is the σ_{V1}, the third column is σ_{U1}...

Value

An object inheriting from class `general.family` of the 'mgcv' package, which can be used in the dsfa package.

References

- Schmidt R, Kneib T (2022). "Multivariate Distributional Stochastic Frontier Models." *arXiv preprint arXiv:2208.10294*.
- Wood SN, Fasiolo M (2017). "A generalized Fellner-Schall method for smoothing parameter optimization with application to Tweedie location, scale and shape models." *Biometrics*, **73**(4), 1071–1081.

Examples

```
#Set seed, sample size and type of function
set.seed(1337)
N=1000 #Sample size
s1<--1 #Set to production function for margin 1
```

```
s2<-1 #Set to cost function for margin 2

#Generate covariates
x1<-runif(N,-1,1); x2<-runif(N,-1,1); x3<-runif(N,-1,1)
x4<-runif(N,-1,1); x5<-runif(N,-1,1); x6<-runif(N,-1,1)
x7<-runif(N,-1,1)

mu1=4+x1 #production function parameter 1
sigma_v1=exp(-1.5+0.75*x2) #noise parameter 1
sigma_u1=exp(-1+1.25*x3) #inefficiency parameter 1
mu2=3+2*x4 #cost function parameter 2
sigma_v2=exp(-1.5+0.75*x5) #noise parameter 2
sigma_u2=exp(-1+.75*x6) #inefficiency parameter 2
Tau<-(exp(1+2.5*x7)-1)/(exp(1+2.5*x7)+1) #Kendall's tau

#Simulate responses and create dataset
Y<-rcomperr_mv(n=N,
               mu1=mu1, sigma_v1=sigma_v1, par_u1 = sigma_u1, s1=s1, dist1="normhnorm",
               mu2=mu2, sigma_v2=sigma_v2, par_u2 = sigma_u2, s2=s2, dist2="normhnorm",
               Tau=Tau, family=1)
dat<-data.frame(y1=Y[,1],y2=Y[,2], x1, x2, x3, x4, x5, x6, x7)

#Write formulae for parameters
mu_1_formula<-y1~x1
sigma_v1_formula<-~x2
sigma_u1_formula<-~x3
mu_2_formula<-y2~x4
sigma_v2_formula<-~x5
sigma_u2_formula<-~x6
tau_formula<-~x7

#Fit model
model<-mgcv::gam(formula=list(mu_1_formula,sigma_v1_formula,sigma_u1_formula,
                              mu_2_formula,sigma_v2_formula,sigma_u2_formula,
                              tau_formula),
            data=dat, family=comperr_mv(s1=s1, s2=s2, dist=c("normhnorm","normhnorm",1)),
                optimizer="efs")

#Model summary
summary(model)
```

---

dcomperr *Composed error term distribution*

Description

Probablitiy density function, distribution, quantile function and random number generation for the composed error term distribution.

Usage

```
dcomperr(
  x = 0,
  mu = 0,
  sigma_v = 1,
  par_u = 1,
  s = -1,
  dist = NULL,
  deriv = 0,
  xg = NULL,
  log.p = FALSE
)

pcomperr(
  q,
  mu = 0,
  sigma_v = 1,
  par_u = 1,
  s = -1,
  dist = NULL,
  deriv = 0,
  xg = NULL,
  lower.tail = TRUE,
  log.p = FALSE
)

qcomperr(
  p,
  mu = 0,
  sigma_v = 1,
  par_u = 1,
  s = -1,
  dist = NULL,
  lower.tail = TRUE,
  log.p = FALSE
)

rcomperr(n, mu = 0, sigma_v = 1, par_u = 1, s = -1, dist = NULL)
```

Arguments

x	vector of quantiles.
mu	vector of μ
sigma_v	vector of σ_V. Must be positive.
par_u	vector of parameter of the (in)efficiency term. Must be positive.
s	$s = -1$ for production and $s = 1$ for cost function.

`dist`	$normhnorm$ for normal-halfnormal and $normexp$ for normal-exponential distribution.
`deriv`	derivative of order `deriv` of the log density. Available are 1,2,3,4.
`xg`	optional, index arrays for upper triangular matrices, generated by `trind.generator(K)` and supplied to `chainrule`.
`log.p`	logical; if TRUE, probabilities p are given as log(p).
`q`	vector of quantiles.
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations.

Details

This is wrapper function for the normal-halfnormal and normal-exponential distribution. A random variable $\mathcal{E}$ follows a composed error distribution if $\mathcal{E} = V + s \cdot U$, where $V \sim N(\mu, \sigma_V^2)$ and $U \sim HN(0, \sigma_U^2)$ or $U \sim Exp(0, \sigma_U^2)$. For more details see `dnormhnorm` and `dnormexp`. Here, $s = -1$ for production and $s = 1$ for cost function.

Value

`dcomperr` gives the density, `pcomperr` give the distribution function, `qcomperr` gives the quantile function, and `rcomperr` generates random numbers, with given parameters. If the derivatives are calculated these are provided as the attributes `gradient`, `hessian`, `l3` and `l4` of the output of the density.

Functions

- `pcomperr()`: distribution function for the composed error distribution.
- `qcomperr()`: quantile function for the composed error distribution.
- `rcomperr()`: random number generation for the composed error distribution.

References

- Aigner D, Lovell CK, Schmidt P (1977). "Formulation and estimation of stochastic frontier production function models." *Journal of econometrics*, **6**(1), 21–37.
- Kumbhakar SC, Wang H, Horncastle AP (2015). *A practitioner's guide to stochastic frontier analysis using Stata*. Cambridge University Press.
- Schmidt R, Kneib T (2020). "Analytic expressions for the Cumulative Distribution Function of the Composed Error Term in Stochastic Frontier Analysis with Truncated Normal and Exponential Inefficiencies." *arXiv preprint arXiv:2006.03459*.
- Gradshteyn IS, Ryzhik IM (2014). *Table of integrals, series, and products*. Academic press.
- Azzalini A (2013). *The skew-normal and related families*, volume 3. Cambridge University Press.

Examples

```
pdf <- dcomperr(x=seq(-3, 3, by=0.1), mu=1, sigma_v=2, par_u=3, s=-1, dist="normhnorm")
cdf <- pcomperr(q=seq(-3, 3, by=0.1), mu=1, sigma_v=2, par_u=3, s=-1, dist="normhnorm")
q <- qcomperr(p=seq(0.1, 0.9, by=0.1), mu=1, sigma_v=2, par_u=3, s=-1, dist="normhnorm")
r <- rcomperr(n=100, mu=1, sigma_v=2, par_u=3, s=-1, dist="normhnorm")
```

`dcomperr_mv` *Composed error multivariate distribution*

Description

Probablitiy density function, distribution and random number generation for the composed error multivariate distribution.

Usage

```
dcomperr_mv(
  x1 = 0,
  mu1 = 0,
  sigma_v1 = 1,
  par_u1 = 1,
  s1 = -1,
  dist1 = "normhnorm",
  x2 = 0,
  mu2 = 0,
  sigma_v2 = 1,
  par_u2 = 1,
  s2 = -1,
  dist2 = "normhnorm",
  Tau = 0,
  family = 1,
  deriv = 0,
  xg = NULL,
  log.p = FALSE
)

pcomperr_mv(
  q1 = 0,
  mu1 = 0,
  sigma_v1 = 1,
  par_u1 = 1,
  s1 = -1,
  dist1 = "normhnorm",
  q2 = 0,
  mu2 = 0,
  sigma_v2 = 1,
```

```
    par_u2 = 1,
    s2 = -1,
    dist2 = "normhnorm",
    Tau = 0,
    family = 1,
    deriv = 0,
    xg = NULL,
    log.p = FALSE
  )

  rcomperr_mv(
    n,
    mu1 = 0,
    sigma_v1 = 1,
    par_u1 = 1,
    s1 = -1,
    dist1 = "normhnorm",
    mu2 = 0,
    sigma_v2 = 1,
    par_u2 = 1,
    s2 = -1,
    dist2 = "normhnorm",
    Tau = 0,
    family = 1
  )
```

Arguments

x1	vector of quantiles for margin 1.
mu1	vector of μ for margin 1
sigma_v1	vector of σ_V for margin 1. Must be positive.
par_u1	vector of σ_U for margin 1. Must be positive.
s1	$s = -1$ for production and $s = 1$ for cost function for margin 1.
dist1	specifying the distribution of margin 1.
x2	vector of quantiles for margin 2.
mu2	vector of μ for margin 2
sigma_v2	vector of σ_V for margin 2. Must be positive.
par_u2	vector of σ_U for margin 2. Must be positive.
s2	$s = -1$ for production and $s = 1$ for cost function for margin 2.
dist2	specifying the distribution of margin 1.
Tau	matrix of Kendall's tau.
family	integer, defines the bivariate copula family: 1 = Gaussian copula 3 = Clayton copula 4 = Gumbel copula

`deriv`	derivative of order `deriv` of the log density. Available are 1,2,3,4.
`xg`	optional, index arrays for upper triangular matrices, generated by `trind.generator(K)` and supplied to `chainrule`.
`log.p`	logical; if TRUE, probabilities p are given as log(p).
`q1`	vector of quantiles for margin 1.
`q2`	vector of quantiles for margin 2.
`n`	number of observations.

Details

A bivariate random vector (Y_1, Y_2) follows a composed error multivariate distribution $f_{Y_1,Y_2}(y_1, y_2)$, which can be rewritten using Sklars' theorem via a copula

$$f_{Y_1,Y_2}(y_1, y_2) = c(F_{Y_1}(y_1), F_{Y_2}(y_2), \tau) \cdot f_{Y_1}(y_1) f_{Y_2}(y_2) \qquad ,$$

where $c(\cdot)$ is a copula function and $F_{Y_m}(y_m)$,$f_{Y_m}(y_m)$ are the marginal cdf and pdf respectively. Tau is Kendall's Tau.

Value

`dcomperr_mv` gives the density and `rcomperr_mv` generates random numbers, with given parameters. If the derivatives are calculated these are provided as the attributes `gradient`, `hessian`, `l3` and `l4` of the output of the density.

Functions

- `pcomperr_mv()`: distribution function for the composed error multivariate distribution.
- `rcomperr_mv()`: random number generation for the composed error multivariate distribution.

References

- Aigner D, Lovell CK, Schmidt P (1977). "Formulation and estimation of stochastic frontier production function models." *Journal of econometrics*, **6**(1), 21–37.

Examples

```
pdf<-dcomperr_mv(x1=5, mu1=2, sigma_v1=3, par_u1=4, s1=-1, dist1="normhnorm",
                 x2=-5, mu2=2, sigma_v2=3, par_u2=4, s2=-1, dist2="normhnorm",
            Tau=0.5, family=1, deriv = 2, xg=NULL, log.p=FALSE)
cdf<-pcomperr_mv(q1=0, mu1=0, sigma_v1=1, par_u1=1, s1=-1, dist1="normhnorm",
            q2=0, mu2=0, sigma_v2=1, par_u2=1, s2=-1, dist2="normhnorm",
            Tau=0.5, family=1, deriv = 0, xg=NULL, log.p=FALSE)
r<-rcomperr_mv(n=100, mu1=0, sigma_v1=1, par_u1=1, s1=-1, dist1="normhnorm",
          mu2=0, sigma_v2=1, par_u2=1, s2=-1, dist2="normhnorm",
          Tau=matrix(0.5,nrow=100), family=1)
```

dcop	*Copula distribution*

Description

Probablity density function and random number generation for the normal, frank and gumbel bivariate copula.

Usage

```
dcop(
  U,
  Tau = 0,
  family = 1,
  deriv = 0,
  disjoint = TRUE,
  num = FALSE,
  log.p = FALSE
)

pcop(U, Tau = 0, family = 1, log.p = FALSE)

rcop(n, Tau = 0, family = 1)
```

Arguments

`U`	matrix of pseudo observations. Must have two columns.
`Tau`	matrix of Kendall's tau.
`family`	integer, defines the copula family: `1` = Gaussian copula
`deriv`	derivative of order `deriv` of the log density. Available are 1,2,3,4.
`disjoint`	logical; if TRUE, only derivatives with respect to Tau are provided.
`num`	logical; if TRUE, numerical derivatives are provided.
`log.p`	logical; if TRUE, probabilities p are given as log(p).
`n`	number of observations.

Details

For more than 2 dimensions only the gaussian copula is implemented. The functions `pcop` and `rcop` are wrapper functions for the `VineCopula` package.

Value

`dcopula` gives the density, `pcop` gives the distribution function for a specified copula and `rcop` generates random numbers, with given Tau. If the derivatives are calculated these are provided as the attributes `gradient`, `hessian`, `l3` and `l4` of the output of the density.

Functions

- pcop(): distribution function for the joint distribution.
- rcop(): random number generation for the joint distribution.

References

- Schepsmeier U, Stöber J (2014). "Derivatives and Fisher information of bivariate copulas." *Statistical Papers*, **55**(2), 525–542.
- Hofert M, Kojadinovic I, Mächler M, Yan J (2018). *Elements of copula modeling with R*. Springer.

Examples

```
pdf <- dcop(U=matrix(c(0.3,0.7), ncol=2), Tau=matrix(0.5,ncol=1), family=1)
cdf <- pcop(U=matrix(c(0.3,0.7), ncol=2), Tau=matrix(0.5,ncol=1), family=1)
r <- rcop(n=100, Tau=matrix(0.5,nrow=100), family=1)
```

---

dnormexp *Normal-exponential distribution*

Description

Probablitiy density function, distribution, quantile function and random number generation for the normal-exponential distribution.

Usage

```
dnormexp(
  x,
  mu = 0,
  sigma_v = 1,
  lambda = 1,
  s = -1,
  deriv = 0,
  xg = NULL,
  log.p = FALSE
)

pnormexp(
  q,
  mu = 0,
  sigma_v = 1,
  lambda = 1,
  s = -1,
  deriv = 0,
```

```
  xg = NULL,
  lower.tail = TRUE,
  log.p = FALSE
)

qnormexp(
  p,
  mu = 0,
  sigma_v = 1,
  lambda = 1,
  s = -1,
  lower.tail = TRUE,
  log.p = FALSE
)

rnormexp(n, mu = 0, sigma_v = 1, lambda = 1, s = -1)
```

Arguments

x	vector of quantiles.
mu	vector of μ
sigma_v	vector of σ_V. Must be positive.
lambda	vector of λ. Must be positive.
s	$s = -1$ for production and $s = 1$ for cost function.
deriv	derivative of order `deriv` of the log density. Available are 1,2,3,4.
xg	optional, index arrays for upper triangular matrices, generated by `trind.generator(K)` and supplied to `chainrule`.
log.p	logical; if TRUE, probabilities p are given as log(p).
q	vector of quantiles.
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
p	vector of probabilities.
n	number of observations.

Details

A random variable $\mathcal{E}$ follows a normal-exponential distribution if $\mathcal{E} = V + s \cdot U$, where $V \sim N(\mu, \sigma_V^2)$ and $U \sim Exp(\lambda)$. The density is given by

$$f_{\mathcal{E}}(\epsilon) = \frac{\lambda}{2} \exp\{\lambda(s\mu) + \frac{1}{2}\lambda^2\sigma_V^2 - \lambda(s\epsilon)\}2\Phi(\frac{1}{\sigma_V}(-s\mu) - \lambda\sigma_V + \frac{1}{\sigma_V}(s\epsilon)) \qquad ,$$

where $s = -1$ for production and $s = 1$ for cost function.

Value

`dnormexp` gives the density, `pnormexp` give the distribution function, `qnormexp` gives the quantile function, and `rnormexp` generates random numbers, with given parameters. If the derivatives are calculated these are provided as the attributes `gradient`, `hessian`, `l3` and `l4` of the output of the density.

Functions

- `pnormexp()`: distribution function for the normal-exponential distribution.
- `qnormexp()`: quantile function for the normal-exponential distribution.
- `rnormexp()`: random number generation for the normal-exponential distribution.

References

- Meeusen W, van Den Broeck J (1977). "Efficiency estimation from Cobb-Douglas production functions with composed error." *International economic review*, 435–444.
- Kumbhakar SC, Wang H, Horncastle AP (2015). *A practitioner's guide to stochastic frontier analysis using Stata*. Cambridge University Press.
- Schmidt R, Kneib T (2020). "Analytic expressions for the Cumulative Distribution Function of the Composed Error Term in Stochastic Frontier Analysis with Truncated Normal and Exponential Inefficiencies." *arXiv preprint arXiv:2006.03459*.
- Gradshteyn IS, Ryzhik IM (2014). *Table of integrals, series, and products*. Academic press.

Examples

```
pdf <- dnormexp(x=seq(-3, 3, by=0.1), mu=1, sigma_v=2, lambda=1/3, s=1)
cdf <- pnormexp(q=seq(-3, 3, by=0.1), mu=1, sigma_v=2, lambda=1/3, s=1)
q <- qnormexp(p=seq(0.1, 0.9, by=0.1), mu=1, sigma_v=2, lambda=1/3, s=1)
r <- rnormexp(n=100, mu=1, sigma_v=2, lambda=1/3, s=1)
```

---

dnormhnorm *Normal-halfnormal distribution*

Description

Probablitiy density function, distribution, quantile function and random number generation for the normal-halfnormal distribution.

Usage

```
dnormhnorm(
  x,
  mu = 0,
  sigma_v = 1,
  sigma_u = 1,
  s = -1,
  deriv = 0,
  xg = NULL,
  log.p = FALSE
)

pnormhnorm(
```

```
    q,
    mu = 0,
    sigma_v = 1,
    sigma_u = 1,
    s = -1,
    deriv = 0,
    xg = NULL,
    lower.tail = TRUE,
    log.p = FALSE
  )

  qnormhnorm(
    p,
    mu = 0,
    sigma_v = 1,
    sigma_u = 1,
    s = -1,
    lower.tail = TRUE,
    log.p = FALSE
  )

  rnormhnorm(n, mu = 0, sigma_v = 1, sigma_u = 1, s = -1)
```

Arguments

x	vector of quantiles.
mu	vector of μ
sigma_v	vector of σ_V. Must be positive.
sigma_u	vector of σ_U. Must be positive.
s	$s = -1$ for production and $s = 1$ for cost function.
deriv	derivative of order `deriv` of the log density. Available are 1,2,3,4.
xg	optional, index arrays for upper triangular matrices, generated by `trind.generator(K)` and supplied to `chainrule`.
log.p	logical; if TRUE, probabilities p are given as log(p).
q	vector of quantiles.
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
p	vector of probabilities.
n	number of observations.

Details

A random variable $\mathcal{E}$ follows a normal-halfnormal distribution if $\mathcal{E} = V + s \cdot U$, where $V \sim N(\mu, \sigma_V^2)$ and $U \sim HN(\sigma_U^2)$. The density is given by

$$f_{\mathcal{E}}(\epsilon) = \frac{1}{\sqrt{\sigma_V^2 + \sigma_U^2}} \phi(\frac{\epsilon - \mu}{\sqrt{\sigma_V^2 + \sigma_U^2}}) \Phi(s \frac{\sigma_U}{\sigma_V} \frac{x - \mu}{\sqrt{\sigma_V^2 + \sigma_U^2}}) \qquad ,$$

where $s = -1$ for production and $s = 1$ for cost function.

Value

dnormhnorm gives the density, pnormhnorm give the distribution function, qnormhnorm gives the quantile function, and rnormhnorm generates random numbers, with given parameters. If the derivatives are calculated these are provided as the attributes gradient, hessian, l3 and l4 of the output of the density.

Functions

- pnormhnorm(): distribution function for the normal-halfnormal distribution.
- qnormhnorm(): quantile function for the normal-halfnormal distribution.
- rnormhnorm(): random number generation for the normal-halfnormal distribution.

References

- Aigner D, Lovell CK, Schmidt P (1977). "Formulation and estimation of stochastic frontier production function models." *Journal of econometrics*, **6**(1), 21–37.
- Kumbhakar SC, Wang H, Horncastle AP (2015). *A practitioner's guide to stochastic frontier analysis using Stata*. Cambridge University Press.
- Schmidt R, Kneib T (2020). "Analytic expressions for the Cumulative Distribution Function of the Composed Error Term in Stochastic Frontier Analysis with Truncated Normal and Exponential Inefficiencies." *arXiv preprint arXiv:2006.03459*.
- Gradshteyn IS, Ryzhik IM (2014). *Table of integrals, series, and products*. Academic press.
- Azzalini A (2013). *The skew-normal and related families*, volume 3. Cambridge University Press.

Examples

```
pdf <- dnormhnorm(x=seq(-3, 3, by=0.1), mu=1, sigma_v=2, sigma_u=3, s=-1)
cdf <- pnormhnorm(q=seq(-3, 3, by=0.1), mu=1, sigma_v=2, sigma_u=3, s=-1)
q <- qnormhnorm(p=seq(0.1, 0.9, by=0.1), mu=1, sigma_v=2, sigma_u=3, s=-1)
r <- rnormhnorm(n=100, mu=1, sigma_v=2, sigma_u=3, s=-1)
```

---

dsfa — *dsfa: Distributional Stochastic Frontier Analysis*

Description

The dsfa package implements the specification, estimation and prediction of distributional stochastic frontier models via mgcv. The basic distributional stochastic frontier model is given by:

$$Y_n = \eta^{\mu}(\boldsymbol{x}_n^{\mu}) + V_n + s \cdot U_n$$

where $n \in \{1, 2, ..., N\}$, V_n and U_n are the noise and (in)efficiency respectively.

- For $s = -1$, $\eta^{\mu}(\cdot)$ is the production function and $\boldsymbol{x}_n^{\mu}$ are the log inputs. Alternatively, if $s = 1$, $\eta^{\mu}(\cdot)$ is the cost function and $\boldsymbol{x}_n^{\mu}$ are the log cost.

- The noise is represented as $V_n \sim N(0, \sigma_{Vn}^2)$, where $\sigma_{Vn} = \exp(\eta^{\sigma_V}(\boldsymbol{x}_n^{\sigma_V}))$. Here, $\boldsymbol{x}_n^{\sigma_V}$ are the observed covariates which influence the parameter of the noise.
- The inefficiency is represented either as $U_n \sim HN(\sigma_{Un}^2)$ or as $U_n \sim Exp(\lambda_n)$, where $\sigma_{Un} = \exp(\eta^{\sigma_{Un}}(\boldsymbol{x}_n^{\sigma_U}))$ and $\lambda_n = \exp(\eta^{\lambda_n}(\boldsymbol{x}_n^{\lambda}))$. Here, $\boldsymbol{x}_n^{\sigma_U}$ or $\boldsymbol{x}_n^{\lambda}$ are the observed covariates which influence the respective parameter of the (in)efficiency.

If $U_n \sim HN(\sigma_{Un}^2)$, then:

$$Y \sim normhnorm(\mu_n = \eta^{\mu}(\boldsymbol{x}_n^{\mu}), \sigma_{Vn} = \exp(\eta^{\sigma_V}(\boldsymbol{x}_n^{\sigma_V})), \sigma_{Un} = \exp(\eta^{\sigma_U}(\boldsymbol{x}_n^{\sigma_U})), s = s)$$

Alternatively, if $U_n \sim Exp(\lambda_n)$ then:

$$Y \sim normexp(\mu_n = \eta^{\mu}(\boldsymbol{x}_n^{\mu}), \sigma_{Vn} = \exp(\eta^{\sigma_V}(\boldsymbol{x}_n^{\sigma_V})), \lambda_n = \exp(\eta^{\lambda}(\boldsymbol{x}_n^{\lambda})), s = s)$$

.

The package fits $\eta^{\mu}(\boldsymbol{x}_n^{\mu}), \eta^{\sigma_V}(\boldsymbol{x}_n^{\sigma_V})$ and $\eta^{\lambda}(\boldsymbol{x}_n^{\lambda})$ or $\eta^{\sigma_U}(\boldsymbol{x}_n^{\sigma_U})$.

Details

The `mgcv` packages provides a framework for fitting distributional regression models. The formulae in `gam` allow for smooth terms utilizing the function `s`. These may be

- linear effects
- non-linear effects which can be modeled via penalized regression splines, e.g. `p.spline`, `tprs`
- random effects, `random.effects`,
- spatial effects which can be modeled via `mrf`.

An overview is provided at `smooth.terms`. The functions `gam`, `predict.gam` and `plot.gam`, are alike to the basic S functions. A number of other functions such as `summary.gam`, `residuals.gam` and `anova.gam` are also provided, for extracting information from a fitted `gamOject`.

The main functions are:

- `normhnorm` Object which can be used to fit a normal-halfnormal stochastic frontier model with the `mgcv` package.
- `normexp` Object which can be used to fit a normal-exponential stochastic frontier model with the `mgcv` package.
- `comperr_mv` Object which can be used to fit a multivariate stochastic frontier model with the `mgcv` package.
- `elasticity` Calculates and plots the elasticity of a smooth function.
- `efficiency` Calculates the expected technical (in)efficiency index $E[u|\epsilon]$ or $E[\exp(-u)|\epsilon]$.

Author(s)

- Rouven Schmidt <rouven.schmidt@tu-clausthal.de>

References

- Schmidt R, Kneib T (2022). "Multivariate Distributional Stochastic Frontier Models." *arXiv preprint arXiv:2208.10294*.
- Wood SN, Fasiolo M (2017). "A generalized Fellner-Schall method for smoothing parameter optimization with application to Tweedie location, scale and shape models." *Biometrics*, **73**(4), 1071–1081.
- Kumbhakar SC, Wang H, Horncastle AP (2015). *A practitioner's guide to stochastic frontier analysis using Stata*. Cambridge University Press.
- Schmidt R, Kneib T (2020). "Analytic expressions for the Cumulative Distribution Function of the Composed Error Term in Stochastic Frontier Analysis with Truncated Normal and Exponential Inefficiencies." *arXiv preprint arXiv:2006.03459*.

`efficiency` *efficiency*

Description

Calculates the expected technical (in)efficiency index.

Usage

```
efficiency(object, level = 0.05, type = "jondrow")
```

Arguments

`object`	fitted mgcv object with family `normhnorm` or `normexp`.
`level`	for the $(1 - level) \cdot 100\%$ confidence interval. Must be in (0,1).
`type`	default is "jondrow" for $E[u\|\epsilon]$, alternatively "battese" for $E[\exp(-u)\|\epsilon]$.

Value

Returns a matrix of the expected (in)efficiency estimates as well the lower and upper bound of the $(1 - level) \cdot 100\%$ confidence interval.

References

- Schmidt R, Kneib T (2022). "Multivariate Distributional Stochastic Frontier Models." *arXiv preprint arXiv:2208.10294*.
- Kumbhakar SC, Wang H, Horncastle AP (2015). *A practitioner's guide to stochastic frontier analysis using Stata*. Cambridge University Press.
- Azzalini A (2013). *The skew-normal and related families*, volume 3. Cambridge University Press.
- Jondrow J, Lovell CK, Materov IS, Schmidt P (1982). "On the estimation of technical inefficiency in the stochastic frontier production function model." *Journal of econometrics*, **19**(2-3), 233–238.
- Battese GE, Coelli TJ (1988). "Prediction of firm-level technical efficiencies with a generalized frontier production function and panel data." *Journal of econometrics*, **38**(3), 387–399.

Examples

```
#Set seed, sample size and type of function
set.seed(1337)
N=500 #Sample size
s=-1 #Set to production function

#Generate covariates
x1<-runif(N,-1,1); x2<-runif(N,-1,1); x3<-runif(N,-1,1)
x4<-runif(N,-1,1); x5<-runif(N,-1,1)

#Set parameters of the distribution
mu=2+0.75*x1+0.4*x2+0.6*x2^2+6*log(x3+2)^(1/4) #production function parameter
sigma_v=exp(-1.5+0.75*x4) #noise parameter
sigma_u=exp(-1+sin(2*pi*x5)) #inefficiency parameter

#Simulate responses and create dataset
y<-rnormhnorm(n=N, mu=mu, sigma_v=sigma_v, sigma_u=sigma_u, s=s)
dat<-data.frame(y, x1, x2, x3, x4, x5)

#Write formulae for parameters
mu_formula<-y~x1+x2+I(x2^2)+s(x3, bs="ps")
sigma_v_formula<-~1+x4
sigma_u_formula<-~1+s(x5, bs="ps")

#Fit model
model<-mgcv::gam(formula=list(mu_formula, sigma_v_formula, sigma_v_formula),
                 data=dat, family=normhnorm(s=s), optimizer = c("efs"))

#Estimate efficiency
efficiency(model, type="jondrow")
efficiency(model, type="battese")
```

`elasticity` *elasticity*

Description

Calculates and plots the elasticity of a smooth function.

Usage

```
elasticity(object, select = NULL, plot = TRUE, se = TRUE)
```

Arguments

`object`	fitted mgcv object with family `normhnorm`, `normexp` or `joint`.

select	specifying the smooth function for which the elasticity is calculated. If `term=NULL` the elasticities for all smooths of μ are returned (excluding random and spatial effects).
plot	logical; if TRUE, plots the elasticities. If FALSE, returns the average elasticity.
se	logical; if TRUE, adds standard errors to the plot of elasticities.

Details

Calculates the marginal product for parametric terms. For smooth terms the average of the derivative is calculated.

Value

If plot is TRUE, plots the elasticities specified in select of the provided object. If plot is FALSE returns a named vector of the elasticity of the provided inputs.

References

- Schmidt R, Kneib T (2022). "Multivariate Distributional Stochastic Frontier Models." *arXiv preprint arXiv:2208.10294*.
- Kumbhakar SC, Wang H, Horncastle AP (2015). *A practitioner's guide to stochastic frontier analysis using Stata*. Cambridge University Press.
- Aigner D, Lovell CK, Schmidt P (1977). "Formulation and estimation of stochastic frontier production function models." *Journal of econometrics*, **6**(1), 21–37.
- Meeusen W, van Den Broeck J (1977). "Efficiency estimation from Cobb-Douglas production functions with composed error." *International economic review*, 435–444.

Examples

```
#Set seed, sample size and type of function
set.seed(1337)
N=500 #Sample size
s=-1 #Set to production function

#Generate covariates
x1<-runif(N,-1,1); x2<-runif(N,-1,1); x3<-runif(N,-1,1)
x4<-runif(N,-1,1); x5<-runif(N,-1,1)

#Set parameters of the distribution
mu=2+0.75*x1+0.4*x2+0.6*x2^2+6*log(x3+2)^(1/4) #production function parameter
sigma_v=exp(-1.5+0.75*x4) #noise parameter
sigma_u=exp(-1+sin(2*pi*x5)) #inefficiency parameter

#Simulate responses and create dataset
y<-rnormhnorm(n=N, mu=mu, sigma_v=sigma_v, sigma_u=sigma_u, s=s)
dat<-data.frame(y, x1, x2, x3, x4, x5)

#Write formulae for parameters
mu_formula<-y~x1+x2+I(x2^2)+s(x3, bs="ps")
```

```
sigma_v_formula<-~1+x4
sigma_u_formula<-~1+s(x5, bs="ps")

#Fit model
model<-mgcv::gam(formula=list(mu_formula, sigma_v_formula, sigma_u_formula),
                 data=dat, family=normhnorm(s=s), optimizer = c("efs"))

#Get elasticities
elasticity(model)
```

---

normexp   *normexp family*

Description

The normexp family implements the normal-exponential distribution in which the μ, σ_V and λ can depend on additive predictors. Useable only with `mgcv::gam`, the additive predictors are specified via a list of formulae.

Usage

```
normexp(link = list("identity", "log", "log"), s = -1)
```

Arguments

`link`	three item list specifying the link for the μ, σ_V and λ parameters. See details.
`s`	$s = -1$ for production and $s = 1$ for cost function.

Details

Used with gam to fit distributional stochastic frontier model. The function `gam` is from the mgcv package is called with a list containing three formulae:

1. The first formula specifies the response on the left hand side and the structure of the additive predictor for μ parameter on the right hand side. Link function is "identity".
2. The second formula is one sided, specifying the additive predictor for the σ_V on the right hand side. Link function is "log".
3. The third formula is one sided, specifying the additive predictor for the λ on the right hand side. Link function is "log".

The fitted values and linear predictors for this family will be three column matrices. The first column is the μ, the second column is the σ_V, the third column is λ.

Value

An object inheriting from class `general.family` of the mgcv package, which can be used in the dsfa package.

References

- Schmidt R, Kneib T (2022). "Multivariate Distributional Stochastic Frontier Models." *arXiv preprint arXiv:2208.10294*.
- Wood SN, Fasiolo M (2017). "A generalized Fellner-Schall method for smoothing parameter optimization with application to Tweedie location, scale and shape models." *Biometrics*, **73**(4), 1071–1081.
- Meeusen W, van Den Broeck J (1977). "Efficiency estimation from Cobb-Douglas production functions with composed error." *International economic review*, 435–444.
- Kumbhakar SC, Wang H, Horncastle AP (2015). *A practitioner's guide to stochastic frontier analysis using Stata*. Cambridge University Press.
- Schmidt R, Kneib T (2020). "Analytic expressions for the Cumulative Distribution Function of the Composed Error Term in Stochastic Frontier Analysis with Truncated Normal and Exponential Inefficiencies." *arXiv preprint arXiv:2006.03459*.

Examples

```
#Set seed, sample size and type of function
set.seed(1337)
N=500 #Sample size
s=-1 #Set to production function

#Generate covariates
x1<-runif(N,-1,1); x2<-runif(N,-1,1); x3<-runif(N,-1,1)
x4<-runif(N,-1,1); x5<-runif(N,-1,1)

#Set parameters of the distribution
mu=2+0.75*x1+0.4*x2+0.6*x2^2+6*log(x3+2)^(1/4) #production function parameter
sigma_v=exp(-1.5+0.75*x4) #noise parameter
lambda=exp(-1+sin(2*pi*x5)) #inefficiency parameter

#Simulate responses and create dataset
y<-rnormexp(n=N, mu=mu, sigma_v=sigma_v, lambda=lambda, s=s)
dat<-data.frame(y, x1, x2, x3, x4, x5)

#Write formulae for parameters
mu_formula<-y~x1+x2+I(x2^2)+s(x3, bs="ps")
sigma_V_formula<-~1+x4
lambda_formula<-~1+s(x5, bs="ps")

#Fit model
model<-mgcv::gam(formula=list(mu_formula, sigma_V_formula, lambda_formula),
                 data=dat, family=normexp(s=s), optimizer = c("efs"))

#Model summary
summary(model)

#Smooth effects
#Effect of x3 on the predictor of the production function
plot(model,select=1) #Estimated function
lines(x3[order(x3)], 6*log(x3[order(x3)]+2)^(1/4)-
```

```
      mean(6*log(x3[order(x3)]+2)^(1/4)),col=2) #True effect

#Effect of x5 on the predictor of the inefficiency
plot(model,select=2) #Estimated function
lines(x5[order(x5)], -1+sin(2*pi*x5)[order(x5)]-
      mean(-1+sin(2*pi*x5)),col=2) #True effect
```

`normhnorm` *normhnorm family*

Description

The normhnorm family implements the normal-halfnormal distribution in which the μ, σ_V and σ_U can depend on additive predictors. Useable only with `mgcv::gam`, the additive predictors are specified via a list of formulae.

Usage

```
normhnorm(link = list("identity", "log", "log"), s = -1)
```

Arguments

`link`	three item list specifying the link for the μ, σ_V and σ_U parameters. See details.
`s`	$s = -1$ for production and $s = 1$ for cost function.

Details

Used with gam to fit distributional stochastic frontier model. The function `gam` is from the mgcv package is called with a list containing three formulae:

1. The first formula specifies the response on the left hand side and the structure of the additive predictor for μ parameter on the right hand side. Link function is "identity".
2. The second formula is one sided, specifying the additive predictor for the σ_V on the right hand side. Link function is "log".
3. The third formula is one sided, specifying the additive predictor for the σ_U on the right hand side. Link function is "log".

The fitted values and linear predictors for this family will be three column matrices. The first column is the μ, the second column is the σ_V, the third column is σ_U.

Value

An object inheriting from class `general.family` of the mgcv package, which can be used in the dsfa package.

References

- Schmidt R, Kneib T (2022). "Multivariate Distributional Stochastic Frontier Models." *arXiv preprint arXiv:2208.10294*.
- Wood SN, Fasiolo M (2017). "A generalized Fellner-Schall method for smoothing parameter optimization with application to Tweedie location, scale and shape models." *Biometrics*, **73**(4), 1071–1081.
- Aigner D, Lovell CK, Schmidt P (1977). "Formulation and estimation of stochastic frontier production function models." *Journal of econometrics*, **6**(1), 21–37.
- Kumbhakar SC, Wang H, Horncastle AP (2015). *A practitioner's guide to stochastic frontier analysis using Stata*. Cambridge University Press.
- Azzalini A (2013). *The skew-normal and related families*, volume 3. Cambridge University Press.
- Schmidt R, Kneib T (2020). "Analytic expressions for the Cumulative Distribution Function of the Composed Error Term in Stochastic Frontier Analysis with Truncated Normal and Exponential Inefficiencies." *arXiv preprint arXiv:2006.03459*.

Examples

```
#Set seed, sample size and type of function
set.seed(1337)
N=500 #Sample size
s=-1 #Set to production function

#Generate covariates
x1<-runif(N,-1,1); x2<-runif(N,-1,1); x3<-runif(N,-1,1)
x4<-runif(N,-1,1); x5<-runif(N,-1,1)

#Set parameters of the distribution
mu=2+0.75*x1+0.4*x2+0.6*x2^2+6*log(x3+2)^(1/4) #production function parameter
sigma_v=exp(-1.5+0.75*x4) #noise parameter
sigma_u=exp(-1+sin(2*pi*x5)) #inefficiency parameter

#Simulate responses and create dataset
y<-rnormhnorm(n=N, mu=mu, sigma_v=sigma_v, sigma_u=sigma_u, s=s)
dat<-data.frame(y, x1, x2, x3, x4, x5)

#Write formulae for parameters
mu_formula<-y~x1+x2+I(x2^2)+s(x3, bs="ps")
sigma_V_formula<-~1+x4
sigma_U_formula<-~1+s(x5, bs="ps")

#Fit model
model<-mgcv::gam(formula=list(mu_formula, sigma_V_formula, sigma_U_formula),
                 data=dat, family=normhnorm(s=s), optimizer = c("efs"))

#Model summary
summary(model)

#Smooth effects
```

```
#Effect of x3 on the predictor of the production function
plot(model, select=1) #Estimated function
lines(x3[order(x3)], 6*log(x3[order(x3)]+2)^(1/4)-
      mean(6*log(x3[order(x3)]+2)^(1/4)), col=2) #True effect

#Effect of x5 on the predictor of the inefficiency
plot(model, select=2) #Estimated function
lines(x5[order(x5)], -1+sin(2*pi*x5)[order(x5)]-
      mean(-1+sin(2*pi*x5)),col=2) #True effect
```

`reparametrize` *reparametrize*

Description

Transforms the given inputs to the parameters and the first three moments of the corresponding distribution. For the normal-halfnormal distribution the parametrization of the classical stochastic frontier as well as the skew-normal and centred skew-normal specification ar provided. For the normal-exponential an the specification via λ and ν are available.

Usage

```
reparametrize(
  mu = NULL,
  sigma_v = NULL,
  sigma_u = NULL,
  s = NULL,
  lambda = NULL,
  nu = NULL,
  xi = NULL,
  omega = NULL,
  alpha = NULL,
  delta = NULL,
  tau = NULL,
  mean = NULL,
  sd = NULL,
  skew = NULL,
  par_u = NULL,
  dist = NULL
)
```

Arguments

mu	vector of μ
sigma_v	vector of σ_V. Must be positive.
sigma_u	vector of σ_U. Must be positive.

s	$s = -1$ for production and $s = 1$ for cost function.
lambda	vector of λ. Must be positive.
nu	vector defined as $\nu = \frac{1}{\lambda}$. Must be positive.
xi	vector of location parameters of the skew-normal distribution defined as $\xi = \mu$
omega	vector of scale parameters of the skew-normal distribution defined as $\omega = \sigma$. Must be positive.
alpha	vector of slant parameters of the skew-normal distribution defined as $\alpha = s\lambda$.
delta	vector of slant parameters rescaled $\frac{\alpha}{\sqrt{1+\alpha^2}}$. Must be within $(-1, 1)$.
tau	vector of the inverted scale parameters of the skew-normal distribution, e.g. $\tau = \frac{1}{\omega}$. Must be positive.
mean	vector of mean of $\mathcal{E}$
sd	vector of standard deviation of $\mathcal{E}$. Must be positive.
skew	vector of skewness of $\mathcal{E}$.
par_u	vector of σ_U or λ. Must be positive.
dist	normhnorm for normal-halfnormal and normexp for normal-exponential distribution.

Details

The following input combinations are allowed for the normal-halfnormal distribution

- mu, sigma_v, sigma_u, s
- xi, omega, alpha
- xi, tau, alpha
- xi, omega, delta
- xi, tau, delta
- mean, sd, skew, dist="normhnorm" ,

while for the normal-exponential distribution the feasible inputs are

- mu, sigma_v, lambda, s
- mu, sigma_v, nu, s
- mean, sd, skew, dist="normexp" .

Other input combinations are not feasible.

Value

Returns a data.frame with the parameter values for all specification.

References

- Kumbhakar SC, Wang H, Horncastle AP (2015). *A practitioner's guide to stochastic frontier analysis using Stata*. Cambridge University Press.
- Azzalini A (2013). *The skew-normal and related families*, volume 3. Cambridge University Press.

Examples

```
#Normal-halfnormal distribution
para<-reparametrize(mu=1, sigma_v=2, sigma_u=3,s=-1)
reparametrize(mean=para$mean, sd=para$sd, skew=para$skew, dist="normhnorm")

#Normal-exponential distribution
para<-reparametrize(mu=1, sigma_v=2, lambda=1/3,s=-1)
reparametrize(mean=para$mean, sd=para$sd, skew=para$skew, dist="normexp")
```

Index