Package 'unifed'

November 6, 2019

Title The Unifed Distribution

Version 1.1.2

Date 2019-11-06

Description Probability functions, family for glm() and Stan code for working with the unifed distribution (Quijano Xacur, 2019; <doi:10.1186/s40488-019-0102-6>).

Depends R (>= 3.1), methods

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests knitr, rmarkdown, testthat, rstan, data.table

VignetteBuilder knitr

Author Oscar Alberto Quijano Xacur [aut,cre]

Maintainer Oscar Alberto Quijano Xacur <oscar.quijano@use.startmail.com>

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-11-06 06:00:02 UTC

R topics documented:

`dirwin.hall` *Irwin-Hall density*

Description

Irwin-Hall density

Usage

```
dirwin.hall(x, n, log = FALSE)
```

Arguments

`x`	A number between 0 and `n`.
`n`	Number of uniform distributions in the unit interval to sum.
`log`	If it evaluates to `TRUE` it returns the log of the density instead of the density.

Details

Gives the density of the Irwin-Hall distribution. It is the density of the sum of `n` uniform distributions on the interval (0,1).

$$h(y;n) = \frac{1}{(n-1)!}\sum_{k=0}^{\lfloor y \rfloor}(-1)^k \binom{n}{k}(y-k)^{n-1}$$

where $x \in [0,1]$ and n is a positive integer.

This function is not numerically stable. The examples have some cases of this.

Examples

```
dirwin.hall(2,5)

# Numerically unstable example
# Run the following one after the other
# See how it goes from positive to negative (which means overflowing )
dirwin.hall(35,50)
dirwin.hall(36,50)
dirwin.hall(37,50)
dirwin.hall(38,50)
```

dunifed *The unifed distribution*

Description

Density, distribution function, quantile function and random generation for the unifed distribution.

Usage

```
dunifed(x, theta)

unifed.lcdf(x, theta)

punifed(q, theta)

qunifed(p, theta)

runifed(n, theta)
```

Arguments

x	A vector of quantiles. They must be numbers between 0 and 1.
theta	The value of the canonical parameter. It must be of length one.
q	A vector of quantiles.
p	A vector of probabilities.
n	number of observations

Value

`dunifed` gives the density function.

`unifed.lcdf` returns the log of the cumulative distribution function of the unifed.

`punifed` gives the distribution function.

`qunifed` gives the quantile function.

`runifed` generates random observations.

References

Quijano Xacur, O.A. The unifed distribution. J Stat Distrib App 6, 13 (2019). doi:10.1186/s40488-019-0102-6.

Examples

```
dunifed( c(0.1,0.3,0.7), 10)


x <- c(0.3,0.6,0.9)
unifed.lcdf(x,5)


x <- c(0.1,0.4,0.7,1)
punifed(x,-5)


p <- 1:9/10
qunifed(p,5)


runifed(20,-3.3)
```

---

summary_unifed_glm *Summarizing Generalized Linear Model Fits*

Description

Wrapper function for summary.glm.

Usage

```
summary_unifed_glm(object, ...)
```

Arguments

object	an object of class "glm".
...	Other arguments for stats::summary.glm.
	This wrapper function was created in order to automatically set to 1 the dispersion parameter of a fitted unifed GLM. When the package is loaded the summary method of the glm class is rewritten using this function.

`unifed` *Family object for the unifed distribution*

Description

Family object for the unifed distribution

Usage

```
unifed(link = "logit", ...)

quasiunifed(link = "logit", ...)

unifed.canonical.link()
```

Arguments

`link`	a specification for the model link function. This can be a name/expression, a literal character string, a length-one character vector or an object of class '"link-glm"' (such as generated by 'make.link') provided it is not specified via one of the accepted names. The `unifed` family accepts the links (as names) 'canonical', 'logit', 'probit', 'cloglog' and 'cauchit'.
`...`	Optional `tol` and `maxit` arguments for `unifed.unit.deviance`.

Details

The link 'canonical' is not part of the standard names accepted by `make.link()` from the stats package. It corresponds to the canonical link function for the unifed distribution, which is the inverse of the derivative of its cumulant generator. There is no explicit formula for it. The function `unifed.kappa.prime.inverse()` implements it using the Newthon-Raphson method.

Value

`unifed` returns a family object for using the unifed distribution with the `glm` function.

The `quasiunifed` family differs from the `unifed` only in that the dispersion parameter is not fixed to one.

An object of class `"link-glm"`.

References

Jørgensen, Bent (1992). The Theory of Exponential Dispersion Models and Analysis of Deviance. Instituto de Matemática Pura e Aplicada, (IMPA), Brazil.

Wedderburn, R. W. M. (1974). Quasi-likelihood functions, generalized linear models, and the Gauss—Newton method. Biometrika. **61** (3): 439–447.

McCullagh, Peter; Nelder, John (1989). Generalized Linear Models (second ed.). London: Chapman and Hall.

See Also

Gamma unifed.kappa.prime.inverse

make.link

---

unifed.deviance *Deviance of the unifed distribution*

Description

Deviance of the unifed distribution

Usage

```
unifed.deviance(y.v, mu.v, wt = 1, ...)

unifed.unit.deviance(y, mu, tol = 1e-07, maxit = 50)
```

Arguments

y.v	A numeric vector with values between 0 and 1
mu.v	A numeric vector with values between 0 and 1
wt	(default value: 1) The weight vector. It contains the weight of each observation. It must contain positive integers only.
...	Additional parameters of unifed.kappa.prime.inverse.one
y	A vector with values between 0 and 1.
mu	A vector with values betwee0 and 1.
tol	Tolerance level for the Newton-Raphson algorithm for computing the inverse of the derivative of the cumulant generator of the family.
maxit	Maximum number of iterations for the Newton-Raphson algorithm for computing the inverse of the derivative of the cumulant generator of the family.

Details

unifed.unit.deviance uses the following expression for the deviance of regular exponential dispersion families

$$d(y,\mu) = 2\left[y\{\dot{\kappa}^{-1}(y) - \dot{\kappa}^{-1}(\mu)\} - \kappa(\dot{\kappa}^{-1}(y)) + \kappa(\dot{\kappa}^{-1}(\mu))\right]$$

$\dot{\kappa}^{-1}$ is computed with the function unifed.kappa.prime.inverse from this package.

Value

`unifed.deviance` returns the deviance of a GLM with a unifed response distribution. This is

$$D(\boldsymbol{y}, \boldsymbol{\mu}) = \sum_{i=1}^{m} w_i d(y_i, \mu_i)$$

Where $d(y_i, \mu_i)$ is the unit deviance of the unifed distribution between the i-th entry of $\boldsymbol{y}$ and $\boldsymbol{\mu}$. w_i is the i-th entry of the weight vector. `unifed.unit.deviance` is used to get the value of d.

`unifed.unit.deviance`

`unifed.kappa` *Cumulant generator of the unifed distribution*

Description

Cumulant generator of the unifed distribution

Usage

```
unifed.kappa(theta)

unifed.kappa.prime(theta)

unifed.kappa.double.prime(theta)

unifed.kappa.prime.inverse(mu, ...)

unifed.kappa.prime.inverse.one(mu, tol = 1e-07, maxit = 1e+07)
```

Arguments

`theta`	A numeric vector.
`mu`	A vector of numbers between 0 and 1
`...`	Other parameters of `unifed.kappa.prime.inverse.one`
`tol`	Tolerance level. The algorithm stops if the proportional difference between the new and old value of an iteration is less or equal than this number.
`maxit`	Maximum number of iterations of the algorithm to look for convergence.

Details

The cumulant generator of the unifed distribution is defined as

$$\kappa(\theta) = \begin{cases} \log\left(\frac{e^{\theta}-1}{\theta}\right) & if \theta \neq 0 \\ 0 & \text{if } \theta = 0 \end{cases} .$$

`unifed.kappa.prime.inverse.one` uses the Newthon-Raphson method for finding the inverse of `unifed.kappa.prime` for a single value.

Value

`unifed.kappa` returns a vector that contains the cumulant generator of the unifed distribution applied to each element of theta.

`unifed.kappa.prime` returns a vector that contains the derivative of the cumulant generator of the unifed distribution for each element of theta.

`unifed.kappa.double.prime` returns a vector that contains the second derivative of the cumulant generator of the unifed distribution for each element of theta.

`unifed.kappa.prime.inverse` returns a vector with `unifed.kappa.prime.inverse.one` evaluated at every entry of `mu`.

`unifed.kappa.prime.inverse.one` if the tolerance level is reached within `maxit` iterations, the function returns the value of the last iteration. Otherwise it returns `NA`.

References

Quijano Xacur, O.A. The unifed distribution. J Stat Distrib App 6, 13 (2019). doi:10.1186/s40488-019-0102-6.

Jørgensen, Bent (1997). The Theory of Dispersion Models. Chapman & Hall, London.

Examples

```
unifed.kappa(1)
unifed.kappa(-5:5)


unifed.kappa.prime(4.5)


unifed.kappa.double.prime(0)


unifed.kappa.prime.inverse(0.5)
unifed.kappa.prime.inverse(c(0.1,0.7,0.9))
```

`unifed.mle` *Maximum Likelihood Estimate for the unifed distribution*

Description

Maximum Likelihood Estimate for the unifed distribution

Usage

```
unifed.mle(x)
```

Arguments

x	A numeric vector with values in the interval [0,1]. Computes the maximum likelihood estimator of the canonical parameter of the unifed distribution. It is assumed that the elements of x come from independent and identically distributed unifed random variables.

Examples

```
a.unifed.sample <- runifed(1000,10)
theta.mle <- unifed.mle(a.unifed.sample)
```

---

unifed.stan *Stan functions for working with the unifed distribution*

Description

Stan functions for working with the unifed distribution

Details

A script with stan functions of the unifed is provided. The script can be included in stan code. The full path to the script can be obtained with the function `unifed.stan.path`. The following list are the names of functions that take one real value:

`real unifed_kappa(real theta)` Computes the cumulant generator of the unifed distribution.

`real unifed_kappa_prime(real theta)` Computes the first derivative of the cumulant generator.

`real unifed_kappa_double_prime(real theta)` Computes the second derivative of the cumulant generator.

`real unifed_lpdf(real x,real theta)` Computes the logarithm of the probability density function of a unifed distribution. `theta` is the value of the canonical parameter of the unifed and `x` if the value where the density is evaluated.

`real unifed_quantile(real p,real theta)` Returns the p-th quantile of a unifed distribution with canonical parameter `theta`.

`real unifed_rng(real theta)` Returns a simulated value of a unifed distribution with canonical parameter `theta`.

`real unifed_lcdf(real x,real theta)` Computes the logarithm of the cumulative density function of a unifed distribution. `theta` is the value of the canonical parameter of the unifed and `x` if the value where the density is evaluated.

`real unifed_kappa_prime_inverse(real mu)` Returns the inverse of the derivative of the unifed cumulant generator

`real unifed_unit_deviance(real y,real mu)` Unit deviance function of the unifed.

The following functions take vectors as arguments

`vector unifed_kappa_v(vector theta)` Vectorized version of `unifed_kappa`.

`vector unifed_kappa_prime_inverse_v(vector mu)` Vectorized version of `unifed_kappa_prime_inverse`.

`void unifed_glm_lp(vector y, vector theta, vector weights)` Adds to the Log Probability Accumulator the logarithm of the likelihood function of a GLM with observed response `y`, estimated canonical parameter `theta` and weights `weights`.

---

unifed.stan.path *Unifed Stan function paths*

Description

The unifed.stan provided by the file contains functions for using the unifed distribution in stan. The file can be included (with #include) insided the functions block of a stan program or its contents can be copied and pasted.

Usage

```
unifed.stan.path()

unifed.stan.folder()
```

Value

The full path to the unifed.stan file provided by the package.

`unifed.stan.folder` returns a string containing the path to the folder containing the unifed.stan file. This can be used as the `isystem` parameter in stan functions.

---

unifed.varf *Variance function of the unifed distribution*

Description

Variance function of the unifed distribution

Usage

```
unifed.varf(mu)
```

Arguments

`mu`	A vector with numbers between 0 and 1.

Value

It returns `unifed.kappa.double.prime(unifed.kappa.prime.inverse(mu))`.

Index