Package 'tweedie'

July 22, 2025

Version 2.3.5

Date 2022-08-17

Title Evaluation of Tweedie Exponential Family Models

Author Peter K. Dunn [cre, aut]

Maintainer Peter K. Dunn <pdunn2@usc.edu.au>

Depends R (>= 2.8.0)

Imports methods

Suggests stabledist, statmod(>= 1.4.20)

Description Maximum likelihood computations for Tweedie families, including the series expansion (Dunn and Smyth, 2005; <doi:10.1007/s11222-005-4070-y>) and the Fourier inversion (Dunn and Smyth, 2008; <doi:10.1007/s11222-007-9039-6>), and related methods.

License GPL (>= 2)

NeedsCompilation yes

RoxygenNote 6.0.1

Repository CRAN

Date/Publication 2022-08-17 07:40:02 UTC

Contents

tweedie-package *Tweedie Distributions*

Description

Functions for computing and fitting the Tweedie family of distributions

Details

Package:	tweedie
Type:	Package
Version:	2.3.2
Date:	2017-12-14
License:	GPL (>=2)

Author(s)

Peter K Dunn

Maintainer: Peter K Dunn <pdunn2@usc.edu.au>

References

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi: 10.1007/s1122200790396

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi: 10.1007/s112220054070y

Dunn, Peter K and Smyth, Gordon K (2001). Tweedie family densities: methods of evaluation. *Proceedings of the 16th International Workshop on Statistical Modelling*, Odense, Denmark, 2–6 July

Jorgensen, B. (1987). Exponential dispersion models. *Journal of the Royal Statistical Society*, B, **49**, 127–162.

Jorgensen, B. (1997). *Theory of Dispersion Models*. Chapman and Hall, London.

Tweedie, M. C. K. (1984). An index which distinguishes between some important exponential families. *Statistics: Applications and New Directions. Proceedings of the Indian Statistical Institute Golden Jubilee International Conference* (Eds. J. K. Ghosh and J. Roy), pp. 579–604. Calcutta: Indian Statistical Institute.

Examples

```
# Generate random numbers
set.seed(987654)
y <- rtweedie( 20, xi=1.5, mu=1, phi=1)
# With Tweedie index  xi   between 1 and 2, this produces continuous
# data with exact zeros
x <- rnorm( length(y), 0, 1)  # Unrelated predictor

# With exact zeros, Tweedie index  xi  must be between 1 and 2

# Fit the tweedie distribution; expect  xi  about 1.5
library(statmod)

xi.vec <- seq(1.1, 1.9, by=0.5)
out <- tweedie.profile( y~1, xi.vec=xi.vec, do.plot=TRUE, verbose=TRUE)

# Fit the glm
require(statmod) # Provides  tweedie  family functions
summary(glm( y ~ x, family=tweedie(var.power=out$xi.max, link.power=0) ))
```

`AICtweedie` *Tweedie Distributions*

Description

The AIC for Tweedie glms

Usage

```
AICtweedie( glm.obj, dispersion=NULL, k = 2, verbose=TRUE)
```

Arguments

`glm.obj`	a fitted Tweedie `glm` object
`dispersion`	the dispersion parameter ϕ; the default is `NULL` which means to use an estimate
`k`	numeric: the penalty per parameter to be used; the default is $k = 2$
`verbose`	if `TRUE` (the default), a warning message is produced about the Poisson case; see the second Note below

Details

See `AIC` for more details on the AIC; see `dtweedie` for more details on computing the Tweedie densities

Value

Returns a numeric value with the corresponding AIC (or BIC, depending on k)

Note

Computing the AIC may take a long time.

Note

Tweedie distributions with the index parameter as 1 correspond to Poisson distributions when $\phi = 1$. However, in general a Tweedie distribution with an index parameter equal to one may not be referring to a Poisson distribution with $\phi = 1$, so we cannot assume that $\phi = 1$ just because the index parameter is set to one. If the Poisson distribution is intended, then `dispersion=1` should be specified. The same argument applies for similar situations.

Author(s)

Peter Dunn (<pdunn2@usc.edu.au>)

References

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi: 10.1007/s1122200790396

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi: 10.1007/s112220054070y

Jorgensen, B. (1997). *Theory of Dispersion Models*. Chapman and Hall, London.

Sakamoto, Y., Ishiguro, M., and Kitagawa G. (1986). *Akaike Information Criterion Statistics*. D. Reidel Publishing Company.

See Also

AIC

Examples

```
library(statmod) # Needed to use  tweedie  family object

### Generate some fictitious data
test.data <- rgamma(n=200, scale=1, shape=1)

### Fit a Tweedie glm and find the AIC
m1 <- glm( test.data~1, family=tweedie(link.power=0, var.power=2) )

### A Tweedie glm with p=2 is equivalent to a gamma glm:
m2 <- glm( test.data~1, family=Gamma(link=log))

### The models are equivalent, so the AIC shoud be the same:
AICtweedie(m1)
AIC(m2)
```

`dtweedie.dldphi` *Tweedie Distributions*

Description

Derivatives of the log-likelihood with respect to ϕ

Usage

```
dtweedie.dldphi(phi, mu, power, y )
dtweedie.dldphi.saddle(phi, mu, power, y )
```

Arguments

`y`	vector of quantiles
`mu`	the mean
`phi`	the dispersion
`power`	the value of p such that the variance is $\text{var}[Y] = \phi\mu^p$

Details

The Tweedie family of distributions belong to the class of exponential dispersion models (EDMs), famous for their role in generalized linear models. The Tweedie distributions are the EDMs with a variance of the form $\text{var}[Y] = \phi\mu^p$ where p is greater than or equal to one, or less than or equal to zero. **This function only evaluates for p greater than or equal to one.** Special cases include the normal ($p = 0$), Poisson ($p = 1$ with $\phi = 1$), gamma ($p = 2$) and inverse Gaussian ($p = 3$) distributions. For other values of `power`, the distributions are still defined but cannot be written in closed form, and hence evaluation is very difficult.

Value

the value of the derivative $\partial\ell/\partial\phi$ where ℓ is the log-likelihood for the specified Tweedie distribution. `dtweedie.dldphi.saddle` uses the saddlepoint approximation to determine the derivative; `dtweedie.dldphi` uses an infinite series expansion.

Author(s)

Peter Dunn (<`pdunn2@usc.edu.au`>)

References

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi: 10.1007/s1122200790396

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi: 10.1007/s112220054070y

Dunn, Peter K and Smyth, Gordon K (2001). Tweedie family densities: methods of evaluation. *Proceedings of the 16th International Workshop on Statistical Modelling*, Odense, Denmark, 2–6 July

Jorgensen, B. (1987). Exponential dispersion models. *Journal of the Royal Statistical Society*, B, **49**, 127–162.

Jorgensen, B. (1997). *Theory of Dispersion Models*. Chapman and Hall, London.

Sidi, Avram (1982). The numerical evaluation of very oscillatory infinite integrals by extrapolation. *Mathematics of Computation* **38**(158), 517–529. doi: 10.1090/S00255718198206456675

Sidi, Avram (1988). A user-friendly extrapolation method for oscillatory infinite integrals. *Mathematics of Computation* **51**(183), 249–266. doi: 10.1090/S00255718198809421535

Tweedie, M. C. K. (1984). An index which distinguishes between some important exponential families. *Statistics: Applications and New Directions. Proceedings of the Indian Statistical Institute Golden Jubilee International Conference* (Eds. J. K. Ghosh and J. Roy), pp. 579-604. Calcutta: Indian Statistical Institute.

See Also

dtweedie.saddle, dtweedie, tweedie.profile, tweedie

Examples

```
### Plot dl/dphi against candidate values of phi
power <- 2
mu <- 1
phi <- seq(2, 8, by=0.1)

set.seed(10000) # For reproducability
y <- rtweedie( 100, mu=mu, power=power, phi=3)
   # So we expect the maximum to occur at  phi=3

dldphi <- dldphi.saddle <- array( dim=length(phi))

for (i in (1:length(phi))) {
   dldphi[i] <- dtweedie.dldphi( y=y, power=power, mu=mu, phi=phi[i])
   dldphi.saddle[i] <- dtweedie.dldphi.saddle( y=y, power=power, mu=mu, phi=phi[i])
}

plot( dldphi ~ phi, lwd=2, type="l",
   ylab=expression(phi), xlab=expression(paste("dl / d",phi) ) )
lines( dldphi.saddle ~ phi, lwd=2, col=2, lty=2)
legend( "bottomright", lwd=c(2,2), lty=c(1,2), col=c(1,2),
   legend=c("'Exact' (using series)","Saddlepoint") )

# Neither are very good in this case!
```

dtweedie.saddle — *Tweedie Distributions (saddlepoint approximation)*

Description

Saddlepoint density for the Tweedie distributions

Usage

```
dtweedie.saddle(y, xi=NULL, mu, phi, eps=1/6, power=NULL)
```

Arguments

`y`	the vector of responses
`xi`	the value of ξ such that the variance is $\text{var}[Y] = \phi\mu^\xi$
`power`	a synonym for ξ
`mu`	the mean
`phi`	the dispersion
`eps`	the offset in computing the variance function. The default is `eps=1/6` (as suggested by Nelder and Pregibon, 1987).

Details

The Tweedie family of distributions belong to the class of exponential dispersion models (EDMs), famous for their role in generalized linear models. The Tweedie distributions are the EDMs with a variance of the form $\text{var}[Y] = \phi\mu^p$ where p is greater than or equal to one, or less than or equal to zero. **This function only evaluates for p greater than or equal to one.** Special cases include the normal ($p = 0$), Poisson ($p = 1$ with $\phi = 1$), gamma ($p = 2$) and inverse Gaussian ($p = 3$) distributions. For other values of `power`, the distributions are still defined but cannot be written in closed form, and hence evaluation is very difficult.

When $1 < p < 2$, the distribution are continuous for Y greater than zero, with a positive mass at $Y = 0$. For $p > 2$, the distributions are continuous for Y greater than zero.

This function approximates the density using the saddlepoint approximation defined by Nelder and Pregibon (1987).

Value

saddlepoint (approximate) density for the given Tweedie distribution with parameters `mu`, `phi` and `power`.

Author(s)

Peter Dunn (<pdunn2@usc.edu.au>)

References

Daniels, H. E. (1954). Saddlepoint approximations in statistics. *Annals of Mathematical Statistics*, **25**(4), 631–650.

Daniels, H. E. (1980). Exact saddlepoint approximations. *Biometrika*, **67**, 59–63. doi: 10.1093/biomet/67.1.59

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi: 10.1007/s1122200790396

Dunn, Peter K and Smyth, Gordon K (2001). Tweedie family densities: methods of evaluation. *Proceedings of the 16th International Workshop on Statistical Modelling*, Odense, Denmark, 2–6 July

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi: 10.1007/s112220054070y

Jorgensen, B. (1987). Exponential dispersion models. *Journal of the Royal Statistical Society*, B, **49**, 127-162.

Jorgensen, B. (1997). *Theory of Dispersion Models*, Chapman and Hall, London.

Nelder, J. A. and Pregibon, D. (1987). An extended quasi-likelihood function. *Biometrika*, **74**(2), 221–232. doi: 10.1093/biomet/74.2.221

Tweedie, M. C. K. (1984). An index which distinguishes between some important exponential families. *Statistics: Applications and New Directions. Proceedings of the Indian Statistical Institute Golden Jubilee International Conference* (Eds. J. K. Ghosh and J. Roy), pp. 579-604. Calcutta: Indian Statistical Institute.

See Also

`dtweedie`

Examples

```
p <- 2.5
mu <- 1
phi <- 1
y <- seq(0, 10, length=100)
fy <- dtweedie( y=y, power=p, mu=mu, phi=phi)
plot(y, fy, type="l")
# Compare to the saddlepoint density
f.saddle <- dtweedie.saddle( y=y, power=p, mu=mu, phi=phi)
lines( y, f.saddle, col=2 )
```

`logLiktweedie` *Tweedie Distributions*

Description

The log likelihood for Tweedie models

Usage

```
logLiktweedie( glm.obj, dispersion=NULL)
```

Arguments

glm.obj	a fitted Tweedie glm object
dispersion	the dispersion parameter ϕ; the default is NULL which means to use an estimate

Details

The log-likelihood is computed from the AIC, so see AICtweedie for more details.

Value

Returns the log-likelihood from the specified model

Note

Computing the log-likelihood may take a long time.

Note

Tweedie distributions with the index parameter as 1 correspond to Poisson distributions when $\phi = 1$. However, in general a Tweedie distribution with an index parameter equal to one may not be referring to a Poisson distribution with $\phi = 1$, so we cannot assume that $\phi = 1$ just because the index parameter is set to one. If the Poisson distribution is intended, then dispersion=1 should be specified. The same argument applies for similar situations.

Author(s)

Peter Dunn (<pdunn2@usc.edu.au>)

References

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi: 10.1007/s1122200790396

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi: 10.1007/s112220054070y

Jorgensen, B. (1997). *Theory of Dispersion Models*. Chapman and Hall, London.

Sakamoto, Y., Ishiguro, M., and Kitagawa G. (1986). *Akaike Information Criterion Statistics*. D. Reidel Publishing Company.

See Also

AICtweedie

Examples

```
library(statmod) # Needed to use  tweedie  family object

### Generate some fictitious data
test.data <- rgamma(n=200, scale=1, shape=1)

### Fit a Tweedie glm and find the AIC
m1 <- glm( test.data~1, family=tweedie(link.power=0, var.power=2) )

### A Tweedie glm with p=2 is equivalent to a gamma glm:
m2 <- glm( test.data~1, family=Gamma(link=log))

### The models are equivalent, so the AIC shoud be the same:
logLiktweedie(m1)
logLik(m2)
```

Tweedie *Tweedie Distributions*

Description

Density, distribution function, quantile function and random generation for the Tweedie family of distributions

Usage

```
dtweedie(y, xi=NULL, mu, phi, power=NULL)
dtweedie.series(y, power, mu, phi)
dtweedie.inversion(y, power, mu, phi, exact=TRUE, method)
dtweedie.stable(y, power, mu, phi)
ptweedie(q, xi=NULL, mu, phi, power=NULL)
ptweedie.series(q, power, mu, phi)
qtweedie(p, xi=NULL, mu, phi, power=NULL)
rtweedie(n, xi=NULL, mu, phi, power=NULL)
```

Arguments

y, q	vector of quantiles
p	vector of probabilities
n	the number of observations
xi	the value of ξ such that the variance is var$[Y] = \phi\mu^\xi$
power	a synonym for ξ
mu	the mean
phi	the dispersion

`exact` logical flag; if `TRUE` (the default), exact zeros are used with the W-algorithm of Sidi (1982); if `FALSE`, approximate (asymptotic) zeros are used in place of exact zeros. Using asymptotic zeros requires less computation but is often less accurate; using exact zeros can be slower but generally improves accuracy.

`method` either `1`, `2` or `3`, determining which of three methods to use to compute the density using the inversion method. If `method` is `NULL` (the default), the optimal method (in terms of relative accuracy) is used, element-by-element of `y`. See the Note in the Details section below

Details

The Tweedie family of distributions belong to the class of exponential dispersion models (EDMs), famous for their role in generalized linear models. The Tweedie distributions are the EDMs with a variance of the form $\text{var}[Y] = \phi\mu^p$ where p is greater than or equal to one, or less than or equal to zero. **This function only evaluates for p greater than or equal to one.** Special cases include the normal ($p = 0$), Poisson ($p = 1$ with $\phi = 1$), gamma ($p = 2$) and inverse Gaussian ($p = 3$) distributions. For other values of `power`, the distributions are still defined but cannot be written in closed form, and hence evaluation is very difficult.

When $1 < p < 2$, the distribution are continuous for Y greater than zero, with a positive mass at $Y = 0$. For $p > 2$, the distributions are continuous for Y greater than zero.

This function evaluates the density or cumulative probability using one of two methods, depending on the combination of parameters. One method is the evaluation of an infinite series. The second interpolates some stored values computed from a Fourier inversion technique.

The function `dtweedie.inversion` evaluates the density using a Fourier series technique; `ptweedie.inversion` does likewise for the cumulative probabilities. The actual code is contained in an external FORTRAN program. Different code is used for $p > 2$ and for $1 < p < 2$.

The function `dtweedie.series` evaluates the density using a series expansion; a different series expansion is used for $p > 2$ and for $1 < p < 2$. The function `ptweedie.series` does likewise for the cumulative probabilities but only for $1 < p < 2$.

The function `dtweedie.stable` exploits the link between the stable distribution (Nolan, 1997) and Tweedie distributions, as discussed in Jorgensen, Chapter 4. These are computed using Nolan's algorithm as implemented in the `stabledist` package (which is therefore required to use the `dtweedie.stable` function).

The function `dtweedie` uses a two-dimensional interpolation procedure to compute the density for some parts of the parameter space from previously computed values found from the series or the inversion. For other parts of the parameter space, the series solution is found.

`ptweedie` returns either the computed series solution or inversion solution.

Value

density (`dtweedie`), probability (`ptweedie`), quantile (`qtweedie`) or random sample (`rtweedie`) for the given Tweedie distribution with parameters `mu`, `phi` and `power`.

Note

The `methods` changed from version 1.4 to 1.5 (methods 1 and 2 swapped). The methods are defined in Dunn and Smyth (2008).

Author(s)

Peter Dunn (<pdunn2@usc.edu.au>)

References

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi: 10.1007/s1122200790396

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi: 10.1007/s112220054070y

Dunn, Peter K and Smyth, Gordon K (2001). Tweedie family densities: methods of evaluation. *Proceedings of the 16th International Workshop on Statistical Modelling*, Odense, Denmark, 2–6 July

Jorgensen, B. (1987). Exponential dispersion models. *Journal of the Royal Statistical Society*, B, **49**, 127–162.

Jorgensen, B. (1997). *Theory of Dispersion Models*. Chapman and Hall, London.

Nolan, John P (1997). Numerical calculation of stable densities and distribution functions. *Communication in Statistics—Stochastic models*, **13**(4). 759–774. doi: 10.1080/15326349708807450

Sidi, Avram (1982). The numerical evaluation of very oscillatory infinite integrals by extrapolation. *Mathematics of Computation* **38**(158), 517–529. doi: 10.1090/S00255718198206456675

Sidi, Avram (1988). A user-friendly extrapolation method for oscillatory infinite integrals. *Mathematics of Computation* **51**(183), 249–266. doi: 10.1090/S00255718198809421535

Tweedie, M. C. K. (1984). An index which distinguishes between some important exponential families. *Statistics: Applications and New Directions. Proceedings of the Indian Statistical Institute Golden Jubilee International Conference* (Eds. J. K. Ghosh and J. Roy), pp. 579-604. Calcutta: Indian Statistical Institute.

See Also

dtweedie.saddle

Examples

```
### Plot a Tweedie density
power <- 2.5
mu <- 1
phi <- 1
y <- seq(0, 6, length=500)
fy <- dtweedie( y=y, power=power, mu=mu, phi=phi)
plot(y, fy, type="l", lwd=2, ylab="Density")
# Compare to the saddlepoint density
f.saddle <- dtweedie.saddle( y=y, power=power, mu=mu, phi=phi)
lines( y, f.saddle, col=2 )
legend("topright", col=c(1,2), lwd=c(2,1),
    legend=c("Actual","Saddlepoint") )

### A histogram of Tweedie random numbers
hist( rtweedie( 1000, power=1.2, mu=1, phi=1) )
```

```
### An example of the multimodal feature of the Tweedie
### family with power near 1 (from Dunn and Smyth, 2005).
y <- seq(0.001,2,len=1000)
mu <- 1
phi <- 0.1
p <- 1.02
f1 <- dtweedie(y,mu=mu,phi=phi,power=p)
plot(y, f1, type="l", xlab="y", ylab="Density")
p <- 1.05
f2<- dtweedie(y,mu=mu,phi=phi,power=p)
lines(y,f2, col=2)

### Compare series and saddlepoint methods
y <- seq(0.001,2,len=1000)
mu <- 1
phi <- 0.1
p <- 1.02
f.series <- dtweedie.series( y,mu=mu,phi=phi,power=p )
f.saddle <- dtweedie.saddle( y,mu=mu,phi=phi,power=p )

f.all <- c( f.series, f.saddle )
plot( range(f.all) ~ range( y ), xlab="y", ylab="Density",
  type="n")
lines( f.series ~ y, lty=1, col=1)
lines( f.saddle ~ y, lty=3, col=3)

legend("topright", lty=c(1,3), col=c(1,3),
  legend=c("Series","Saddlepoint") )
```

`Tweedie internals` *Tweedie internal function*

Description

Internal tweedie functions.

Usage

```
dtweedie.dlogfdphi(y, mu, phi, power)
dtweedie.logl(phi, y, mu, power)
dtweedie.logl.saddle( phi, power, y, mu, eps=0)
dtweedie.logv.bigp( y, phi, power)
dtweedie.logw.smallp(y, phi, power)
dtweedie.interp(grid, nx, np, xix.lo, xix.hi,p.lo, p.hi, power, xix)
dtweedie.jw.smallp(y, phi, power )
dtweedie.kv.bigp(y, phi, power)
dtweedie.series.bigp(power, y, mu, phi)
```

```
dtweedie.series.smallp(power, y, mu, phi)
stored.grids(power)
twpdf(p, phi, y, mu, exact, verbose, funvalue, exitstatus, relerr, its )
twcdf(p, phi, y, mu, exact,          funvalue, exitstatus, relerr, its )
```

Arguments

`y`	the vector of responses
`power`	the value of p such that the variance is $\text{var}[Y] = \phi\mu^p$
`mu`	the mean
`phi`	the dispersion
`grid`	the interpolation grid necessary for the given value of p
`nx`	the number of interpolation points in the ξ dimension
`np`	the number of interpolation points in the p dimension
`xix.lo`	the lower value of the transformed ξ value used in the interpolation grid. (Note that the value of ξ is from 0 to ∞, and is transformed such that it is on the range 0 to 1.)
`xix.hi`	the higher value of the transformed ξ value used in the interpolation grid.
`p.lo`	the lower value of p value used in the interpolation grid.
`p.hi`	the higher value of p value used in the interpolation grid.
`xix`	the value of the transformed ξ at which a value is sought.
`eps`	the offset in computing the variance function in the saddlepoint approximation. The default is `eps=1/6` (as suggested by Nelder and Pregibon, 1987).
`p`	the Tweedie index parameter
`exact`	a flag for the FORTRAN to use exact-zeros acceleration algorithmic the calculation (1 means to do so)
`verbose`	a flag for the FORTRAN: 1 means to be verbose
`funvalue`	the value of the call returned by the FORTRAN code
`exitstatus`	the exit status returned by the FORTRAN code
`relerr`	an estimation of the relative error returned by the FORTRAN code
`its`	the number of iterations of the algorithm returned by the FORTRAN code

Details

These are not to be called by the user.

Author(s)

Peter Dunn (<`pdunn2@usc.edu.au`>)

References

Nelder, J. A. and Pregibon, D. (1987). An extended quasi-likelihood function *Biometrika*, **74**(2), 221–232. doi10.1093/biomet/74.2.221

`tweedie.convert` *Convert Tweedie parameters*

Description

Converts Tweedie distribution parameters to the parameters of the underlying distributions

Usage

```
tweedie.convert( xi=NULL, mu, phi, power=NULL)
```

Arguments

`xi`	the value of ξ such that the variance is $\text{var}[Y] = \phi\mu^\xi$
`power`	a synonym for ξ
`mu`	the mean
`phi`	the dispersion

Details

The Tweedie family of distributions with $1 < \xi < 2$ is the Poisson sum of gamma distributions (where the Poisson distribution has mean λ, and the gamma distribution has scale and shape parameters). When used to fit a glm, the model is fitted with the usual glm parameters: the mean μ and the dispersion parameter ϕ. This function converts the parameters (p, μ, ϕ) to the values of the parameters of the underlying Poisson distribution λ and gamma distribution (scale and shape parameters).

Value

a list containing the values of the mean of the underlying Poisson distribution (as `poisson.lambda`), the scale parameter of the underlying gamma distribution (as `gamma.scale`), the shape parameter of the underlying gamma distribution (as `gamma.shape`), the probability of obtaining a zero response (as `p0`), the mean of the underlying gamma distribution (as `gamma.mean`), and the dispersion parameter of the underlying gamma distribution (as `gamma.phi`).

Author(s)

Peter Dunn (<pdunn2@usc.edu.au>)

References

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi: 10.1007/s1122200790396

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi: 10.1007/s112220054070y

Dunn, Peter K and Smyth, Gordon K (2001). Tweedie family densities: methods of evaluation. *Proceedings of the 16th International Workshop on Statistical Modelling*, Odense, Denmark, 2–6 July

Tweedie, M. C. K. (1984). An index which distinguishes between some important exponential families. *Statistics: Applications and New Directions. Proceedings of the Indian Statistical Institute Golden Jubilee International Conference* (Eds. J. K. Ghosh and J. Roy), pp. 579-604. Calcutta: Indian Statistical Institute.

See Also

`dtweedie.saddle`

Examples

```
tweedie.convert(xi=1.5, mu=1, phi=1)
```

`tweedie.dev` *Tweedie Distributions: the deviance function*

Description

The deviance function for the Tweedie family of distributions

Usage

```
tweedie.dev(y, mu, power)
```

Arguments

`y`	vector of quantiles (which can be zero if $1 < p < 2$
`mu`	the mean
`power`	the value of p such that the variance is $\text{var}[Y] = \phi\mu^p$

Details

The Tweedie family of distributions belong to the class of exponential dispersion models (EDMs), famous for their role in generalized linear models. The Tweedie distributions are the EDMs with a variance of the form $\text{var}[Y] = \phi\mu^p$ where p is greater than or equal to one, or less than or equal to zero. **This function only evaluates for p greater than or equal to one.** Special cases include the normal ($p = 0$), Poisson ($p = 1$ with $\phi = 1$), gamma ($p = 2$) and inverse Gaussian ($p = 3$) distributions. For other values of `power`, the distributions are still defined but cannot be written in closed form, and hence evaluation is very difficult.

The deviance is defined by `deviance` as "up to a constant, minus twice the maximized log-likelihood. Where sensible, the constant is chosen so that a saturated model has deviance zero."

Value

the value of the deviance for the given Tweedie distribution with parameters `mu`, `phi` and `power`.

Author(s)

Peter Dunn (<pdunn2@usc.edu.au>)

References

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi: 10.1007/s1122200790396

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi: 10.1007/s112220054070y

Dunn, Peter K and Smyth, Gordon K (2001). Tweedie family densities: methods of evaluation. *Proceedings of the 16th International Workshop on Statistical Modelling*, Odense, Denmark, 2–6 July

Jorgensen, B. (1987). Exponential dispersion models. *Journal of the Royal Statistical Society*, B, **49**, 127–162.

Jorgensen, B. (1997). *Theory of Dispersion Models*. Chapman and Hall, London.

Sidi, Avram (1982). The numerical evaluation of very oscillatory infinite integrals by extrapolation. *Mathematics of Computation* **38**(158), 517–529. doi: 10.1090/S00255718198206456675

Sidi, Avram (1988). A user-friendly extrapolation method for oscillatory infinite integrals. *Mathematics of Computation* **51**(183), 249–266. doi: 10.1090/S00255718198809421535

Tweedie, M. C. K. (1984). An index which distinguishes between some important exponential families. *Statistics: Applications and New Directions. Proceedings of the Indian Statistical Institute Golden Jubilee International Conference* (Eds. J. K. Ghosh and J. Roy), pp. 579-604. Calcutta: Indian Statistical Institute.

See Also

dtweedie, dtweedie.saddle, tweedie, deviance, glm

Examples

```
### Plot a Tweedie deviance function when 1<p<2
mu <- 1

y <- seq(0, 6, length=100)

dev1 <- tweedie.dev( y=y, mu=mu, power=1.1)
dev2 <- tweedie.dev( y=y, mu=mu, power=1.5)
dev3 <- tweedie.dev( y=y, mu=mu, power=1.9)

plot(range(y), range( c(dev1, dev2, dev3)),
   type="n", lwd=2, ylab="Deviance", xlab=expression(italic(y)) )

lines( y, dev1, lty=1, col=1, lwd=2 )
lines( y, dev2, lty=2, col=2, lwd=2 )
lines( y, dev3, lty=3, col=3, lwd=2 )


legend("top", col=c(1,2,3), lwd=c(2,2,2), lty=c(1,2,3),
```

```
    legend=c("p=1.1","p=1.5", "p=1.9") )


### Plot a Tweedie deviance function when p>2
mu <- 1

y <- seq(0.1, 6, length=100)

dev1 <- tweedie.dev( y=y, mu=mu, power=2) # Gamma
dev2 <- tweedie.dev( y=y, mu=mu, power=3) # Inverse Gaussian
dev3 <- tweedie.dev( y=y, mu=mu, power=4)

plot(range(y), range( c(dev1, dev2, dev3)),
   type="n", lwd=2, ylab="Deviance", xlab=expression(italic(y)) )

lines( y, dev1, lty=1, col=1, lwd=2 )
lines( y, dev2, lty=2, col=2, lwd=2 )
lines( y, dev3, lty=3, col=3, lwd=2 )


legend("top", col=c(1,2,3), lwd=c(2,2,2), lty=c(1,2,3),
    legend=c("p=2 (gamma)", "p=3 (inverse Gaussian)", "p=4") )
```

`tweedie.plot` *Tweedie Distributions: plotting*

Description

Plotting Tweedie density and distribution functions

Usage

```
tweedie.plot(y, xi, mu, phi, type="pdf", power=NULL, add=FALSE, ...)
```

Arguments

`y`	vector of values at which to evaluate and plot
`xi`	the value of ξ such that the variance is $\text{var}[Y] = \phi\mu^\xi$
`power`	a synonym for ξ
`mu`	the mean
`phi`	the dispersion
`type`	what to plot: `pdf` (the default) means the probability function, or `cdf`, the cumulative distribution function
`add`	if `TRUE`, the plot is added to the current device; if `FALSE` (the default), a new plot is produced
`...`	Arguments to be passed to the plotting method

Details

For details, see dtweedie

Value

this function is usually called for side-effect of producing a plot of the specified Tweedie distribution, properly plotting the exact zero that occurs at $y = 0$ when $1 < p < 2$. However, it also produces a list with the computed density at the given points, with components y and x respectively, such that `plot(y~x)` approximately reproduces the plot.

Author(s)

Peter Dunn (<pdunn2@usc.edu.au>)

References

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi: 10.1007/s1122200790396

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi: 10.1007/s112220054070y

Dunn, Peter K and Smyth, Gordon K (2001). Tweedie family densities: methods of evaluation. *Proceedings of the 16th International Workshop on Statistical Modelling*, Odense, Denmark, 2–6 July

Jorgensen, B. (1987). Exponential dispersion models. *Journal of the Royal Statistical Society*, B, **49**, 127–162.

Jorgensen, B. (1997). *Theory of Dispersion Models*. Chapman and Hall, London.

Nolan, John P (1997). Numerical calculation of stable densities and distribution functions. *Communication in Statistics—Stochastic models*, **13**(4). 759–774. doi: 10.1080/15326349708807450

Sidi, Avram (1982). The numerical evaluation of very oscillatory infinite integrals by extrapolation. *Mathematics of Computation* **38**(158), 517–529. doi: 10.1090/S00255718198206456675

Sidi, Avram (1988). A user-friendly extrapolation method for oscillatory infinite integrals. *Mathematics of Computation* **51**(183), 249–266. doi: 10.1090/S00255718198809421535

Tweedie, M. C. K. (1984). An index which distinguishes between some important exponential families. *Statistics: Applications and New Directions. Proceedings of the Indian Statistical Institute Golden Jubilee International Conference* (Eds. J. K. Ghosh and J. Roy), pp. 579-604. Calcutta: Indian Statistical Institute.

See Also

dtweedie

Examples

```
### Plot a Tweedie density with 1<p<2
yy <- seq(0,5,length=100)
tweedie.plot( power=1.7, mu=1, phi=1, y=yy, lwd=2)
tweedie.plot( power=1.2, mu=1, phi=1, y=yy, add=TRUE, lwd=2, col="red")
```

```
legend("topright",lwd=c(2,2), col=c("black","red"), pch=c(19,19),
   legend=c("p=1.7","p=1.2") )

### Plot distribution functions
tweedie.plot( power=1.05, mu=1, phi=1, y=yy,
   lwd=2, type="cdf", ylim=c(0,1))
tweedie.plot( power=2, mu=1, phi=1, y=yy,
   add=TRUE, lwd=2, type="cdf",col="red")
legend("bottomright",lwd=c(2,2), col=c("black","red"),
   legend=c("p=1.05","p=2") )

### Now, plot two densities, combining p>2 and 1<p<2
tweedie.plot( power=3.5, mu=1, phi=1, y=yy, lwd=2)
tweedie.plot( power=1.5, mu=1, phi=1, y=yy, lwd=2, col="red", add=TRUE)
legend("topright",lwd=c(2,2), col=c("black","red"), pch=c(NA,19),
   legend=c("p=3.5","p=1.5") )
```

---

tweedie.profile *Tweedie Distributions: mle estimation of p*

Description

Maximum likelihood estimation of the Tweedie index parameter p.

Usage

```
tweedie.profile(formula, p.vec=NULL, xi.vec=NULL, link.power=0,
      data, weights, offset, fit.glm=FALSE,
      do.smooth=TRUE, do.plot=FALSE, do.ci=do.smooth,
      eps=1/6,
    control=list( epsilon=1e-09, maxit=glm.control()$maxit, trace=glm.control()$trace ),
      do.points=do.plot, method="inversion", conf.level=0.95,
      phi.method=ifelse(method == "saddlepoint", "saddlepoint", "mle"),
      verbose=FALSE, add0=FALSE)
```

Arguments

`formula`	a formula expression as for other regression models and generalized linear models, of the form `response ~ predictors`. For details, see the documentation for `lm`, `glm` and `formula`
`p.vec`	a vector of `p` values for consideration. The values must all be larger than one (if the response variable has exact zeros, the values must all be between one and two). If `NULL` (the default), `p.vec` is set to `seq(1.2, 1.8, by=0.1)` if the response contains any zeros, or `seq(1.5, 5, by=0.5)` if the response contains no zeros. See the DETAILS section below for further details.
`xi.vec`	the same as `p.vec`; some authors use the p notation for the index parameter, and some use ξ; this function detects which is used and then uses that notation throughout

link.power
: the power link function to use. These link functions $g(\cdot)$ are of the form $g(\eta) = \eta^{\text{link.power}}$, and the special case of link.power=0 (the default) refers to the logarithm link function. See the documentation for tweedie also.

data
: an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which glm is called.

weights
: an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector.

offset
: this can be used to specify an *a priori* known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector of length either one or equal to the number of cases. One or more offset terms can be included in the formula instead or as well, and if both are specified their sum is used. See model.offset.

fit.glm
: logical flag. If TRUE, the Tweedie generalized linear model is fitted using the value of p found by the profiling function. If FALSE (the default), no model is fitted.

do.smooth
: logical flag. If TRUE (the default), a spline is fitted to the data to smooth the profile likelihood plot. If FALSE, no smoothing is used (and the function is quicker). **Note** that p.vec must contain *at least five points* for smoothing to be allowed.

do.plot
: logical flag. If TRUE, a plot of the profile likelihood is produce. If FALSE (the default), no plot is produced.

do.ci
: logical flag. If TRUE, the nominal 100*conf.level is computed. If FALSE, no confidence interval is computed. By default, do.ci is the same value as do.smooth, since a confidence interval will only be accurate if smoothing has been performed. Indeed, if do.smooth=FALSE, confidence intervals are never computed and do.ci is forced to FALSE if it is given as TRUE.

eps
: the offset in computing the variance function. The default is eps=1/6 (as suggested by Nelder and Pregibon, 1987). Note eps is ignored unless the method="saddlepoint" as it makes no sense otherwise.

control
: a list of parameters for controlling the fitting process; see glm.control and glm. The default is to use the maximum number of iterations maxit and the trace setting as given in glm.control, but to set epsilon to 1e-09 to ensure a smoother plot

do.points
: plot the points on the plot where the (log-) likelihood is computed for the given values of p; defaults to the same value as do.plot

method
: the method for computing the (log-) likelihood. One of "series", "inversion" (the default), "interpolation" or "saddlepoint". If there are any troubles using this function, sometimes a change of method will fix the problem. Note that method="saddlepoint" is only an approximate method for computing the (log-) likelihood. Using method="interpolation" may produce a jump in the profile likelihood as it changes computational regimes.

conf.level
: the confidence level for the computation of the nominal confidence interval. The default is conf.level=0.95.

phi.method
: the method for estimating phi, one of "saddlepoint" or "mle". A maximum likelihood estimate is used unless method="saddlepoint", when the saddlepoint approximation method is used. Note that using phi.method="saddlepoint" is equivalent to using the mean deviance estimator of phi.

verbose
: the amount of feedback requested: 0 or FALSE means minimal feedback (the default), 1 or TRUE means some feedback, or 2 means to show all feedback. Since the function can be slow and sometimes problematic, feedback can be good; but it can also be unnecessary when one knows all is well.

add0
: if TRUE, the value p=0 is used in forming the profile log-likelihood (corresponding to the normal distribution); the default value is add0=FALSE

Details

For each value in p.vec, the function computes an estimate of phi and then computes the value of the log-likelihood for these parameters. The plot of the log-likelihood against p.vec allows the maximum likelihood value of p to be found. Once the value of p is found, the distribution within the class of Tweedie distribution is identified.

Value

The main purpose of the function is to estimate the value of the Tweedie index parameter, p, which is produced by the output list as p.max. Optionally (if do.plot=TRUE), a plot is produced that shows the profile log-likelihood computed at each value in p.vec (smoothed if do.smooth=TRUE). This function can be temperamental (for theoretical reasons involved in numerically computing the density), and this plot shows the values of p requested on the horizontal axis (using rug); there may be fewer points on the plot, since the likelihood some values of p requested may have returned NaN, Inf or NA.

A list containing the components: y and x (such that plot(x,y) (partially) recreates the profile likelihood plot); ht (the height of the nominal confidence interval); L (the estimate of the (log-) likelihood at each given value of p); p (the p-values used); phi (the computed values of phi at the values in p); p.max (the estimate of the mle of p); L.max (the estimate of the (log-) likelihood at p.max); phi.max (the estimate of phi at p.max); ci (the lower and upper limits of the confidence interval for p); method (the method used for estimation: series, inversion, interpolation or saddlepoint); phi.method (the method used for estimation of phi: saddlepoint or phi).

If glm.fit is TRUE, the list also contains a component glm.obj, a glm object for the fitted Tweedie generalized linear model.

Note

The estimates of p and phi are printed. The result is printed invisibly.

If the response variable has any exact zeros, the values in p.vec must all be between one and two.

The function is sometimes unstable and may fail. It may also be very slow. One solution is to change the method. The default is method="inversion" (the default); then try method="series", method="interpolation" and method="saddlepoint" in that order. Note that method="saddlepoint" is an approximate method only. Also make sure the values in p.vec are suitable for the data (see above paragraph).

It is recommended that for the first use with a data set, use `p.vec` with only a small number of values and set `do.smooth=FALSE`, `do.ci=FALSE`. If this is successful, a larger vector `p.vec` and smoothing can be used.

Author(s)

Peter Dunn (<pdunn2@usc.edu.au>)

References

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi: 10.1007/s1122200790396

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi: 10.1007/s112220054070y

Dunn, Peter K and Smyth, Gordon K (2001). Tweedie family densities: methods of evaluation. *Proceedings of the 16th International Workshop on Statistical Modelling*, Odense, Denmark, 2–6 July

Jorgensen, B. (1987). Exponential dispersion models. *Journal of the Royal Statistical Society*, B, **49**, 127–162.

Jorgensen, B. (1997). *Theory of Dispersion Models*. Chapman and Hall, London.

Nelder, J. A. and Pregibon, D. (1987). An extended quasi-likelihood function. *Biometrika* **74**(2), 221–232. doi: 10.1093/biomet/74.2.221

Tweedie, M. C. K. (1984). An index which distinguishes between some important exponential families. *Statistics: Applications and New Directions. Proceedings of the Indian Statistical Institute Golden Jubilee International Conference* (Eds. J. K. Ghosh and J. Roy), pp. 579-604. Calcutta: Indian Statistical Institute.

See Also

`dtweedie`, `dtweedie.saddle`, `tweedie`

Examples

```
library(statmod) # Needed to use  tweedie.profile
# Generate some fictitious data
test.data <- rgamma(n=200, scale=1, shape=1)
# The gamma is a Tweedie distribution with power=2;
# let's see if p=2 is suggested by  tweedie.profile:
## Not run:
out <- tweedie.profile( test.data ~ 1,
p.vec=seq(1.5, 2.5, by=0.2) )
out$p.max
out$ci

## End(Not run)
```

Index