Package 'TBEinf'

April 28, 2024

Type Package

Title TBEinf

Version 0.0.1

Author Ziyu Xiang [aut,cre], Xu Liu [aut], Shaul Bar-Lev [aut], Ad Ridder [aut]

Maintainer Ziyu Xiang <xiangziyu2867@163.com>

Description Computations for the density of TBE infinity, including direct calculation, saddlepoint approximation, Fourier inversion, and related methods. Estimation and prediction obtained by applying TBE infinity in the generalized linear model.

License GPL (>=2)

Imports pracma

Repository github

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

R topics documented:

`dTBEinf` *Density of TBE infinity*

Description

Computations for the density of TBE_∞, including direct calculation, saddlepoint approximation, Fourier inversion, and modified W-Transformation.

Usage

```
dTBEinf(y, X, beta, phi, method="saddle", m = NULL)
```

Arguments

`y`	Vector of responses $\mathbf{y} \in \mathbb{R}^n$.
`X`	The design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$.
`beta`	Parameter vector $\boldsymbol{\beta}$.
`phi`	The dispersion φ.
`method`	Different methods for calculating density. method="real" uses real density, "saddle" uses saddlepoint approximation, "finverse" uses Fourier inversion, "mW-trans" uses modified W-transformation. Default is "saddle".
`m`	Vector of means. Default is **m**=NULL in which **m** is calculated by $m_i = -\log(-\mathbf{X}_{i\cdot}^\top \boldsymbol{\beta})$. If **m** is not NULL, then **X** and $\boldsymbol{\beta}$ are not used.

Details

The TBE$_\infty$ distribution is the exponential dispersion model generated by the Landau distribution and also known as a Tweedie model with power infinity. Tweedie models are exponential dispersion models possessing a power variance function (VF) of the form $V(m) = m^\gamma, \gamma \in (-\infty, 0] \cup [1, \infty)$, where m is the mean. And TBE$_\infty$ possesses an exponential variance function e^m, which can be considered as a limiting case of the power variance functions of Tweedie.

The density of TBE$_\infty$ is defined but cannot be written in closed form, and hence evaluation is very difficult. Therefore, some approximations are generally used.

Value

A vector with the same dimension as y. The value of the density of TBE$_\infty$ calculated by different methods.

References

Bar-Lev, S.K. (2023). The Exponential Dispersion Model Generated by the Landau Distribution—A Comprehensive Review and Further Developments. *Mathematics* **11**(20), 4343. doi:10.3390/math11204343

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, P. K. and Smyth, G. K. (2018). *Generalized Linear Models with Examples in R*. Springer: Berlin/Heidelberg, Germany.

Dunn, P. K. and Smyth, G. K. (2022). Evaluation of Tweedie Exponential Family Models; R Package Version 2.3.5; 2022. Available online: https://cran.r-project.org/web/packages/tweedie/tweedie.pdf

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

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

Examples

```
y <- seq(10, 20, by = 0.1)
n <- length(y)
p <- 10
x <- matrix(runif(n * p), nrow = n, ncol = p)
X <- cbind(1, x)
beta <- rep(-0.01, p)
m0 <- rep(5,n)
phi <- 0.5
dTBEinf(y, X, beta, phi, method="real")
dTBEinf(y, X, beta, phi)
dTBEinf(y, X, beta, phi, method="finverse", m=m0)
dTBEinf(y, X, beta, phi, method="mWtrans")
```

`logLikTBEinf` *log-likelihood of TBE infinity Distributions*

Description

The log-likelihood for TBE_∞

Usage

```
logLikTBEinf(y, X, beta, phi, m = NULL)
```

Arguments

`y`	Vector of responses $\mathbf{y} \in \mathbb{R}^n$.
`X`	The design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$.
`beta`	Parameter vector $\boldsymbol{\beta}$.
`phi`	The dispersion φ.
`m`	Vector of means. Default is **m**=NULL in which **m** is calculated by $m_i = -\log(-\mathbf{X}_{i\cdot}^\top \boldsymbol{\beta})$. If **m** is not NULL, then **X** and $\boldsymbol{\beta}$ are not used.

Details

`logLikTBEinf` calculates the log-likelihood.

Value

The log-likelihood for TBE_∞. A scalar.

References

Bar-Lev, S.K. (2023). The Exponential Dispersion Model Generated by the Landau Distribution—A Comprehensive Review and Further Developments. *Mathematics* **11**(20), 4343. doi:10.3390/math11204343

Examples

```
y <- seq(10, 20, by = 0.1)
n <- length(y)
p <- 10
x <- matrix(runif(n * p), nrow = n, ncol = p)
X <- cbind(1, x)
beta <- rep(-0.01, p)
m0 <- rep(5,n)
phi <- 0.5
logLikTBEinf(y, X, beta, phi)
logLikTBEinf(y, X, beta, phi, m=m0)
```

TBEinf.Dev *Deviance of TBE infinity Distributions*

Description

The total deviance for TBE_∞

Usage

```
TBEinf.Dev(y, X, beta, m = NULL)
```

Arguments

y	Vector of responses $\mathbf{y} \in \mathbb{R}^n$.
X	The design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$.
beta	Parameter vector $\boldsymbol{\beta}$.
m	Vector of means. Default is **m**=NULL in which **m** is calculated by $m_i = -\log(-\mathbf{X}_{i\cdot}^\top \boldsymbol{\beta})$. If **m** is not NULL, then **X** and $\boldsymbol{\beta}$ are not used.

Details

`TBEinf.Dev` calculates the total deviance.

Value

The total deviance for TBE_∞. A scalar.

References

Bar-Lev, S.K. (2023). The Exponential Dispersion Model Generated by the Landau Distribution —A Comprehensive Review and Further Developments. *Mathematics* **11**(20), 4343. doi:10.3390/math11204343

Dunn, P. K. and Smyth, G. K. (2018). *Generalized Linear Models with Examples in R*. Springer: Berlin/Heidelberg, Germany.

Examples

```
y <- seq(10, 20, by = 0.1)
n <- length(y)
p <- 10
x <- matrix(runif(n * p), nrow = n, ncol = p)
X <- cbind(1, x)
beta <- rep(-0.01, p)
m0 <- rep(5,n)
TBEinf.Dev(y, X, beta)
TBEinf.Dev(y, X, beta, m=m0)
```

`TBEinf.glm.fit` *TBE infinity applied to GLM*

Description

TBE_∞ is applied to the GLM for estimation.

Usage

```
TBEinf.glm.fit(y, X, beta0 = NULL, tau = 1e-8, max.iter = 100)
```

Arguments

`y`	Vector of responses $\mathbf{y} \in \mathbb{R}^n$.
`X`	The design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$.
`beta0`	Initial parameter vector $\boldsymbol{\beta}_0$. Default is $\boldsymbol{\beta}_0$=NULL in which $\boldsymbol{\beta}_0$ is given by $\boldsymbol{\beta}_0 = (-0.01, -0.01, \cdots, -0.01)^\top$.
`tau`	The threshold τ for judging convergence. Default is 1e-8.
`max.iter`	The maximum number of outer-loop iterations allowed. Default is 100.

Details

The TBE_∞ GLM assumes a systematic component where the linear predictor $\eta = \beta_0 + \sum_{j=1}^{p} \beta_j x_j$ is linked to the mean m through a link function $\eta = -e^{-m}$. The estimation of $\boldsymbol{\beta}$ is obtained through Iteratively Reweighted Least Squares (IRLS). The estimate of φ is obtained through mean deviance.

Value

A list.

- beta - Estimate of coefficients of covariates.
- iter - Iterations.
- m - Estimate of mean obtained through link function.
- phi - Estimate of dispersion.

References

Dunn, P. K. and Smyth, G. K. (2018). *Generalized Linear Models with Examples in R*. Springer: Berlin/Heidelberg, Germany.

See Also

TBEinf.Dev

Examples

```
set.seed(123) # For reproducibility
link_inverse <- function(eta) -log(- eta) # Inverse link function

n <- 200
p <- 9
pp <- p+1
beta_0 <- -0.1*(1:pp)
phi_0 <- 0.01

x <- matrix(runif(n * p), nrow = n, ncol = p)
X <- cbind(1, x)
eta_0 <- X
m_0 <- link_inverse(eta_0)
y <- rnorm(n, mean = m_0, sd = phi_0*exp(m_0))

initial <- rep(-0.01, pp)
beta <- TBEinf.glm.fit(y, X, initial)$beta
iter <- TBEinf.glm.fit(y, X, initial)$iter
m <- TBEinf.glm.fit(y, X)$m
phi <- TBEinf.glm.fit(y, X)$phi
```

TBEinf.glm.pre *TBE infinity applied to GLM*

Description

TBE_∞ is applied to the GLM for prediction.

Usage

```
TBEinf.glm.pre(X, beta)
```

Arguments

X	new data x
beta	Parameter vector $\boldsymbol{\beta}$. Generally estimated values.

Details

`TBEinf.glm.pre` predicts y by $\hat{y} = -\log(-\mathbf{x}^\top \beta)$.

Value

The prediction of response y.

Examples

```
set.seed(123) # For reproducibility
link_inverse <- function(eta) -log(- eta) # Inverse link function

n <- 800
p <- 9
pp <- p+1
beta_0 <- -0.1*(1:pp)
phi_0 <- 0.01

x <- matrix(runif(n * p), nrow = n, ncol = p)
X <- cbind(1, x)
eta_0 <- X
m_0 <- link_inverse(eta_0)
y <- rnorm(n, mean = m_0, sd = phi_0*exp(m_0))

beta <- TBEinf.glm.fit(y, X)$beta
phi <- TBEinf.glm.fit(y, X)$phi

x_new <- runif(pp)
y_pre <- TBEinf.glm.pre(x_new, beta)
```

Index