Package 'CompGLM'

April 29, 2018

Type Package

Title Conway-Maxwell-Poisson GLM and Distribution Functions

Version 2.0

Date 2018-04-29

Author Jeffrey Pollock

Maintainer Jeffrey Pollock <jeffpollock9@gmail.com>

URL https://github.com/jeffpollock9/CompGLM

BugReports https://github.com/jeffpollock9/CompGLM/issues

Description A function (which uses a similar interface to the `glm' function) for the fitting of a Conway-Maxwell-Poisson GLM. There are also various methods for analysis of the model fit. The package also contains functions for the Conway-Maxwell-Poisson distribution in a similar interface to functions `dpois', `ppois' and `rpois'. The functions are generally quick, since the workhorse functions are written in C++ (thanks to the Rcpp package).

License GPL (>= 2)

Imports Rcpp, stats

LinkingTo Rcpp

Suggests testthat, devtools, roxygen2, covr, lintr

NeedsCompilation yes

Encoding UTF-8

RoxygenNote 6.0.1

Repository CRAN

Date/Publication 2018-04-29 12:45:36 UTC

R topics documented:

`dcomp` *Conway-Maxwell-Poisson Probability Density Function*

Description

The PDF of the Conway-Maxwell-Poisson distribution with parameters `lam` and `nu` at point `y`.

Usage

```
dcomp(y, lam, nu, sumTo = 100L, logP = FALSE)
```

Arguments

`y`	an integer vector where the density is to be calculated. If the input is not an integer, it will be coerced to be an integer.
`lam`	a double vector of the parameter λ.
`nu`	a double vector of the parameter ν.
`sumTo`	an integer for the summation term in the density (default 100).
`logP`	a boolean for if the log of the density should be given (default `FALSE`).

Details

The Conway-Maxwell-Poisson distribution has density:

$$P(y) = \frac{\lambda^y}{(y!)^\nu Z(\lambda, \nu)}$$

for $y = 0, 1, 2, \ldots$, and:

$$Z(\lambda, \nu) = \sum_{j=0}^{\infty} \frac{\lambda^j}{(j!)^\nu}$$

where the summation is approximated by summing from `j = 0` to `sumTo`.

Value

The value of the PDF (or log PDF if `logP = TRUE`). Input vectors are recycled to be same length.

Author(s)

Jeffrey Pollock <jeffpollock9@gmail.com>

Examples

```
dcomp(-5:5, 2.5, 1)
dcomp(2, 2.5, 1.5)
require(graphics)
require(stats)
comp <- dcomp(0:10, 2.5, 0.9)
poisson <- dpois(0:10, 2.5)
barplot(rbind(comp, poisson), beside = TRUE, names= 0:10, legend.text = TRUE)
```

`glm.comp` *Conway-Maxwell Poisson GLM Fitting Function*

Description

A function in similar format to `glm` which provides a linear form regressing on the parameters lambda and mu.

Usage

```
glm.comp(lamFormula, nuFormula = NULL, data, lamStart = NULL,
  nuStart = NULL, sumTo = 100L, method = "BFGS", ...)
```

Arguments

`lamFormula`	an object of class `formula` which determines the form of regression for the model parameter λ. An offset can also be added in the formula.
`nuFormula`	an object of class `formula` which determines the form of regression for the model parameter ν. The default value is `NULL` meaning the formula is intercept only. An offset can also be added in the formula.
`data`	an optional `data.frame` containing the variables in the model. If not found in data, the variables are taken from `environment(lamFormula)`.
`lamStart`	optional vector of starting values for the coefficients of the λ regression.
`nuStart`	optional vector of starting values for the coefficients of the ν regression.
`sumTo`	an integer for the summation term in the density (default 100).
`method`	optimisation method passed to `optim` (default "BFGS").
`...`	further arguments to be passed to `optim`.

Details

A log link is used for regression of the model parameters λ and ν, that is:

$$\log(\lambda) = \beta X$$

$$\log(\nu) = \zeta Y$$

where: β is the vector of coefficients for the parameter λ, ζ is the vector of coefficients for the parameter ν, X is the model matrix for the parameter λ, and Y is the model matrix for the parameter ν.

The parameter vectors are calculated via maximum likelihood using the general optimisation function optim. A Poisson model will be fit using glm.fit and (unless starting values are supplied) the coefficients will be used as starting values for the parameter vector β.

Several S3 functions have been implemented for model analysis print, coef, extractAIC, logLik, predict, and summary,

Value

An object of class 'Comp' which is a list with all the components needed for the relevant S3 class methods.

Author(s)

Jeffrey Pollock <jeffpollock9@gmail.com>

References

A Flexible Regression Model for Count Data, by Sellers & Shmueli, http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1127359

Examples

```
set.seed(1)
n <- 5000
x1 <- rnorm(n, -1.0, 0.5)
x2 <- rnorm(n, 1.0, 0.7)
x3 <- rnorm(n, 2.0, 0.4)
y <- rpois(n, exp(-0.5 + 0.3 * x1 + 0.8 * x2 + 0.2 * x3))
data <- data.frame(y, x1, x2, x3)
model <- glm.comp(y ~ ., data = data)
print(model)
summary(model)
coef(model)
head(predict(model))
AIC(model)
```

---

pcomp *Conway-Maxwell-Poisson Cumulative Density Function*

Description

The CDF of the Conway-Maxwell-Poisson distribution with parameters lam and nu at point q.

Usage

```
pcomp(q, lam, nu, sumTo = 100L, lowerTail = TRUE, logP = FALSE)
```

Arguments

q	an integer vector where the CDF is to be calculated. If the input is not an integer, it will be coerced to be an integer.
lam	a double vector of the parameter λ.
nu	a double vector of the parameter ν.
sumTo	an integer for the summation term in the density (default 100)
lowerTail	a boolean for if P(Y<=q) should be returned (default TRUE), otherwise, P(Y>q) is returned.
logP	a boolean for if the log of the probability should be given (default FALSE)

Details

See dcomp for details of the PDF.

Value

The value of the CDF (or 1-CDF if `lowerTail = FALSE`) or log of this value if `logP = TRUE`. Input vectors are recycled to be same length.

Author(s)

Jeffrey Pollock <jeffpollock9@gmail.com>

Examples

```
pcomp(1:10, 1.5, 1.2)
```

---

rcomp *Conway-Maxwell-Poisson Random Sample*

Description

Provides a size n sample from the Conway-Maxwell-Poisson distribution with parameters `lam` and `nu`. Sampling is done via a simple multinomial approach.

Usage

```
rcomp(n, lam, nu, sumTo = 100L)
```

Arguments

n	an integer of the number of random samples to be taken.
lam	a double of the parameter λ.
nu	a double of the parameter ν.
sumTo	an integer for the summation term in the density (default 100).

Details

The function is only implemented for single values of `lam` and `nu`. See `dcomp` for details of the PDF.

Value

A random sample of size n.

Author(s)

Jeffrey Pollock <jeffpollock9@gmail.com>

Examples

```
require(graphics)
sample <- rcomp(1000, 8.5, 0.9)
barplot(table(sample))
```

Index