Package 'compoisson'

July 2, 2014

Type Package

Title Conway-Maxwell-Poisson Distribution

Version 0.3

Date 2008-05-07

Author Jeffrey Dunn

Maintainer Jeffrey Dunn <jsd115@gmail.com>

Description Provides routines for density and moments of the Conway-Maxwell-Poisson distribution as well as functions for fitting the COM-Poisson model for over/under-dispersed count data.

License BSD

Depends stats, MASS

Repository CRAN

Date/Publication 2012-10-29 08:58:27

NeedsCompilation no

R topics documented:

`compoisson-package` *Conway-Maxwell Poisson Distribution*

Description

Provides routines for computing the density of the Conway-Maxwell Poisson distribution and fitting parameters to data.

Details

Package:	compoisson
Type:	Package
Version:	0.2
Date:	2008-04-21
License:	BSD

Author(s)

Jeffrey Dunn

Maintainer: Jeffrey Dunn <jsd115@gmail.com>

References

Shmueli, G., Minka, T. P., Kadane, J. B., Borle, S. and Boatwright, P., "A useful distribution for fitting discrete data: Revival of the Conway-Maxwell-Poisson distribution," J. Royal Statist. Soc., v54, pp. 127-142, 2005.

See Also

See `dcom` for calculating the pmf of the distribution, see `com.fit` for fitting parameters.

`com.compute.z` *Compute COM-Poisson Normalizing Constant*

Description

Computes the normalizing constant in the COM-Poisson model for given values of the parameters.

Usage

```
com.compute.z(lambda, nu, log.error = 0.001)
com.compute.log.z(lambda, nu, log.error = 0.001)
```

Arguments

lambda	Lambda value in COM-Poisson distribution
nu	Nu value in COM-Poisson distribution
log.error	Precision in the log of the normalizing constant

Details

com.compute.z computes the COM-Poisson normalizing constant

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

to the specified precision. If no precision is specified, then the package default is used.

com.compute.log.z is equivalent to log(com.compute.z(lambda, nu)) but provudes additional precision.

Value

The normalizing constant as a real number with specified precision.

Author(s)

Jeffrey Dunn

References

Shmueli, G., Minka, T. P., Kadane, J. B., Borle, S. and Boatwright, P., "A useful distribution for fitting discrete data: Revival of the Conway-Maxwell-Poisson distribution," J. Royal Statist. Soc., v54, pp. 127-142, 2005.

See Also

com.fit

Examples

```
data(insurance);
fit = com.fit(Lemaire);
z = com.compute.z(fit$lambda, fit$nu);
```

com.confint *Computes a confidence interval for parameter estimates of the COM-Poisson Distribution*

Description

Computes a pivotal bootstrap confidence interval for maximum likelihood parameter estimates.

Usage

```
com.confint(data, level=0.95, B=1000, n=1000)
```

Arguments

data	the matrix of data to fit
level	the level of the confidence interval
B	number of repetitions of the bootstrap
n	number of data points in each bootstrap sample

Details

Uses a standard pivotal confidence interval from a bootstrap sample.

Value

A matrix containing the confidence intervals for each parameter

Author(s)

Akshaya Jha, Jeffrey Dunn

References

Wasserman, L. (2005). "All of Statistics: A Concise Course in Statistical Inference," Springer Texts in Statistics.

See Also

com.fit

`com.expectation` *Computes Expectation of a Function of a COM-Poisson Random Variable*

Description

Computes an expectation of a function of a COM-Poisson random variable.

Usage

```
com.expectation(f, lambda, nu, log.error = 0.001)
```

Arguments

`f`	function taking as a single argument the value of x
`lambda`	value of lambda parameter
`nu`	value of nu parameter
`log.error`	precision in the log of the expectation

Details

Computes the expectation $E[f(X)]$ where X is a COM-Poisson random variable.

Value

The expectation as a real number.

Author(s)

Jeffrey Dunn

References

Shmueli, G., Minka, T. P., Kadane, J. B., Borle, S. and Boatwright, P., "A useful distribution for fitting discrete data: Revival of the Conway-Maxwell-Poisson distribution," J. Royal Statist. Soc., v54, pp. 127-142, 2005.

See Also

`com.mean`, `com.var`, `com.fit`

com.fit *Computes COM-Poisson Regression*

Description

Computes the maximum likelihood estimates of the COM-Poisson model for given count data.

Usage

```
com.fit(x)
```

Arguments

x	matrix of count data

Details

The argument x should consist of a matrix where the first column is the level and the second column is the count for the corresponding level.

Value

Returns an object containing four fields:

lambda	Estimate of the lambda parameter
nu	Estimate of the nu parameter
z	Normalizing constant
fitted.values	Estimated counts at given levels

Author(s)

Jeffrey Dunn

References

Shmueli, G., Minka, T. P., Kadane, J. B., Borle, S. and Boatwright, P., "A useful distribution for fitting discrete data: Revival of the Conway-Maxwell-Poisson distribution," J. Royal Statist. Soc., v54, pp. 127-142, 2005.

See Also

com.compute.z, com.loglikelihood

Examples

```
data(insurance)
com.fit(Lemaire);
```

com.log.density *Computes the Log PMF of the COM-Poisson Distribution*

Description

Computes the log probability mass function of the COM-Poisson distribution for given values of the parameters.

Usage

```
com.log.density(x, lambda, nu, log.z = NULL)
```

Arguments

x	level to evaulate the log PMF at
lambda	value of the lambda parameter
nu	value of the nu parameter
log.z	log of the normalizing constant, computed if not specified

Details

Computes the log probability mass function of the COM-Poisson distribution

$$\log f(x) = x \log \lambda - \log(Z(\lambda, \nu)) - \nu \sum_{i=1}^{x} x.$$

Value

The log probability that a random COM-Poisson variable X takes value x.

Author(s)

Jeffrey Dunn

References

Shmueli, G., Minka, T. P., Kadane, J. B., Borle, S. and Boatwright, P., "A useful distribution for fitting discrete data: Revival of the Conway-Maxwell-Poisson distribution," J. Royal Statist. Soc., v54, pp. 127-142, 2005.

See Also

com.loglikelihood, dcom

Examples

```
data(insurance);
fit = com.fit(Lemaire);
com.log.density(0, fit$lambda, fit$nu, fit$z);
```

`com.loglikelihood` *Computes Log-Likelihood of COM-Poisson*

Description

Given a set of data, computes the log-likelihood of the data under the COM-Poisson distribution for values of the parameters.

Usage

```
com.loglikelihood(x, lambda, nu)
```

Arguments

`x`	matrix of count data
`lambda`	value of lambda parameter
`nu`	value of nu parameter

Details

The argument x should consist of a matrix where the first column is the level and the second column is the count for the corresponding level.

Value

The log-likelihood of the data.

Author(s)

Jeffrey Dunn

References

Shmueli, G., Minka, T. P., Kadane, J. B., Borle, S. and Boatwright, P., "A useful distribution for fitting discrete data: Revival of the Conway-Maxwell-Poisson distribution," J. Royal Statist. Soc., v54, pp. 127-142, 2005.

See Also

`com.fit`, `dcom`

com.logspace *Operations in Log-space*

Description

Computes the difference of two values in log-space.

Usage

```
com.log.difference(x, y)
com.log.sum(x, y)
com.log.factorial(x)
```

Arguments

x	first value
y	second value

Details

com.log.difference computes the difference of two values in log-space, $log(e^x - e^y)$, without significant chance of overflow or underflow.

com.log.sum computes the sum of two values in log-space, $log(e^x + e^y)$, without significant change of overflow or underflow.

com.log.factorial computes $log(x!)$ which is equivalent to a summation.

Value

The requested computation in log-space.

Author(s)

Jeffrey Dunn

Examples

```
a = exp(com.log.difference(log(100), log(20))); # a = 80
b = exp(com.log.sum(log(100), log(20))); # b = 120
c = exp(com.log.factorial(4)); # c = 24
```

com.mean *Computes Mean of the COM-Poisson Distribution*

Description

Computes the mean of the COM-Poisson distribution for given values of the parameters.

Usage

```
com.mean(lambda, nu)
```

Arguments

lambda	value of lambda parameter
nu	value of the nu parameter

Details

Uses com.expectation to compute the first moment of the distribution.

Value

The mean of the distribution.

Author(s)

Jeffrey Dunn

References

Shmueli, G., Minka, T. P., Kadane, J. B., Borle, S. and Boatwright, P., "A useful distribution for fitting discrete data: Revival of the Conway-Maxwell-Poisson distribution," J. Royal Statist. Soc., v54, pp. 127-142, 2005.

See Also

com.expectation, com.var

Examples

```
data(insurance)
model = com.fit(Lemaire);
com.mean(model$lambda, model$nu);
```

com.var *Computes Variance of the COM-Poisson Distribution*

Description

Computes the variance of the COM-Poisson distribution for given values of the parameters.

Usage

```
com.var(lambda, nu)
```

Arguments

lambda	value of lambda parameter
nu	value of the nu parameter

Details

Uses com.expectation to compute the second moment of the distribution and subtracts the squared mean, computed using com.mean.

Value

The variance of the distribution.

Author(s)

Jeffrey Dunn

References

Shmueli, G., Minka, T. P., Kadane, J. B., Borle, S. and Boatwright, P., "A useful distribution for fitting discrete data: Revival of the Conway-Maxwell-Poisson distribution," J. Royal Statist. Soc., v54, pp. 127-142, 2005.

See Also

com.expectation, com.mean

Examples

```
data(insurance)
model = com.fit(Lemaire);
com.var(model$lambda, model$nu);
```

compoisson-data *Insurance Count Datasets*

Description

Two auto insurance datasets compiled from published works. The Lemaire dataset contains published aggregate claim numbers for automobile third-party liability insurance of a Belgian insurance company in the early 1990's. The Buhlmann dataset originates from aggregate accident claims in 1961 for a class of auto insurance in Switzerland.

Usage

```
data(insurance)
```

Format

Each dataset is a matrix with two columns. The first column contains the levels and the second contains the number of customers who submitted the corresponding level of claims.

Source

Lemaire, Jean. "Bonus-Malus Systems for Automobile Insurance". Kluwer Academic Publishers, 1995.

Panjer, Harry. "Actuarial Mathematics (Proceedings of Symposia in Applied Mathematics)". Providence: American Mathematical Society, 1986.

Examples

```
data(insurance)
Lemaire
Buhlmann
```

dcom *The COM-Poisson Distribution*

Description

Probability mass function and random generation for the COM-Poisson distribution for given values of the parameters.

Usage

```
dcom(x, lambda, nu, z = NULL)
rcom(n, lambda, nu, log.z = NULL)
```

Arguments

x	level to evaluate the PMF at
lambda	value of lambda parameter
nu	value of nu parameter
z	normalizing constant, computed if not specified
n	number of random values to return
log.z	natural log of z

Details

Computes the probability mass function of the COM-Poisson distribution

$$f(x) = \frac{1}{Z(\lambda, \nu)} \frac{\lambda^x}{(x!)^\nu}.$$

Value

dcom gives the probability that a random COM-Poisson variable X takes value x. rcom gives a vector of n random values sampled from the COM-Poisson distribution.

Author(s)

Jeffrey Dunn

References

Shmueli, G., Minka, T. P., Kadane, J. B., Borle, S. and Boatwright, P., "A useful distribution for fitting discrete data: Revival of the Conway-Maxwell-Poisson distribution," J. Royal Statist. Soc., v54, pp. 127-142, 2005.

See Also

com.loglikelihood, com.log.density

Examples

```
data(insurance);
fit = com.fit(Lemaire);
dcom(0, fit$lambda, fit$nu, fit$z);
r = rcom(10, fit$lambda, fit$nu);
```

Index