Package ‘GammaModel’

July 26, 2018

Title Modelling age-at-death profiles using the Gamma distribution

Version 1.0

Maintainer Adrian Timpson <a.timpson@ucl.ac.uk>

Description
Tools to model age-at-death count data using the Gamma distribution. Accompanies 'Modelling caprine age-at-death profiles using the Gamma distribution' by Adrian Timpson, Rosalind E. Gillis, Katie Manning and Mark G. Thomas. Journal of Archaeological Science 2018.

Imports dplyr, combinat, DEoptimR, stats, LaplacesDemon

Depends R (>= 3.1.0)

Suggests scales, knitr, ks

License GPL-3

Encoding UTF-8

VignetteBuilder knitr

NeedsCompilation no

Author Adrian Timpson [aut, cre]

R topics documented:

ageClassLogMLE *GammaModel::ageClassLogMLE*

Description

Finds the Maximum Likelihood of an age-class model, given some age-class count data.

Usage

```
ageClassLogMLE(counts, N = 200, I = 40, C = 2)
```

Arguments

counts	A one-row data frame comprising integer counts, with column names in capital letters. Formatting requirements detailed in checkCountsFormat.
N	An integer specifying the population size of particles in each iteration.
I	An integer specifying the number of iterations.
C	A numeric value greater than 1, average parameter jump size.

Details

A search for the MLE is constrained by the fact we are searching for a vector of model probabilities which must by definition sum to 1. Therefore our general approach is to sample from the n-simplex using the Dirichlet distribution. We use a Random Search Algorithm which samples a population of N particles from the Simplex, and retains the particle with the greatest likelihood. At each subsequent iteration, a new population of N particles is sampled from a region of the simplex close to the previous best particle (also including the previous best particle). At each iteration, the average distance to the previous best particle is reduced, the rate of which is controlled by C.

Value

A single numeric, giving the Maximum Likelihood Estimate.

References

Rastrigin, L.A. (1963). The convergence of the random search method in the extremal control of a many parameter system. Automation and Remote Control. 24 (10): 1337-1342

See Also

ageClassMLparameters

Examples

```
# Maximum Likelihood Age-class parameters of 'TRA1' data
# using the default Payne caprine age classes.
data(Neolithic)
counts <- Neolithic['TRA1',]
ageClassLogMLE(counts)
```

ageClassMLparameters *GammaModel::ageClassMLparameters*

Description

Finds the Maximum Likelihood Age-class parameters given some age-class count data.

Usage

```
ageClassMLparameters(counts, N = 200, I = 40, C = 2)
```

Arguments

`counts`	A one-row data frame comprising integer counts, with column names in capital letters. Formatting requirements detailed in `checkCountsFormat`.
`N`	An integer specifying the population size of particles in each iteration.
`I`	An integer specifying the number of iterations.
`C`	A numeric value greater than 1, average parameter jump size.

Details

A search for the MLE is constrained by the fact we are searching for a vector of model probabilities which must by definition sum to 1. Therefore our general approach is to sample from the n-simplex using the Dirichlet distribution. We use a Random Search Algorithm which samples a population of N particles from the Simplex, and retains the particle with the greatest likelihood. At each subsequent iteration, a new population of N particles is sampled from a region of the simplex close to the previous best particle (also including the previous best particle). At each iteration, the average distance to the previous best particle is reduced, the rate of which is controlled by C.

Value

A two value numeric vector, giving the estimated parameter values.

References

Rastrigin, L.A. (1963). "The convergence of the random search method in the extremal control of a many parameter system. Automation and Remote Control. 24 (10): 1337-1342

See Also

`ageClassLogMLE`

Examples

```
    # Maximum Likelihood Age-class parameters of 'TRA1' data
    # using the default Payne caprine age classes.
    data(Neolithic)
    counts <- Neolithic['TRA1',]
    ageClassMLparameters(counts)
```

allArrangements *GammaModel::allArrangements*

Description

Generates a data frame of all possible arrangements of the counts in rows. Age classes that always share counts remain as multi-classes.

Usage

```
allArrangements(counts)
```

Arguments

counts	A one-row data frame comprising integer counts, with column names in capital letters. Formatting requirements detailed in checkCountsFormat.

Details

Calculating the probability of the observed counts under some model probabilities requires calculating the sum of the probabilities across all possible arrangements (each row).

Memory demands from the total number of arrangements rapidly becomes huge if there are many counts across many multi-classes, rendering the calculation of exact probabilities inpractical.

Value

Data frame of all possible arrangements of the counts in rows, and age classes in columns.

Examples

```
# All 30 possible arrangements for site 'PPI'
# Notice classes E & F (also H & I) always share counts
data(Neolithic)
counts <- Neolithic['PPI',]
allArrangements(counts)

# Number of possible arrangements can be very large
counts <- Neolithic['TRA2',]
x <- allArrangements(counts)
nrow(x)
```

checkCountsFormat *GammaModel::checkCountsFormat*

Description

Helper function to check count data are in the required format

Usage

```
checkCountsFormat(counts)
```

Arguments

counts	A one-row data.frame comprising integer counts, with column names in capital letters

Details

Performs a few basic checks to ensure the raw count data are formatted as required:

A one-row data frame comprising column names of age classes (or multi-classes) in capital letters from 'A'. For example, Payne's 1973 caprine age classes are 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', and 'I'. However, it is more likely that the raw counts will include some multi-class assignments, for example 'ABCD', 'A', 'AB', 'B', 'BC' etc. Order does not matter.

Row names are unconstrained. In the 'Neolithic' dataset provided, these specify the site codes.

Raw counts must by definition be integers.

Deviations from these formatting requirements are output as warnings.

Value

The string 'OK' is returned if no errors are found. Otherwise a string stating the number of errors.

References

Kill-off Patterns in Sheep and Goats: The Mandibles from Asvan Kale. Payne 1973.

Examples

```
# load the raw data counts for all ten Neolithic sites
data(Neolithic)

# select the 'TRA1' subset
counts <- Neolithic['TRA1',]

# eyeball data
print(counts)

# check format
checkCountsFormat(counts)

# change to a non-integer count
mistake <- counts
mistake[3] <- 2.5
```

```
checkCountsFormat(mistake)
```

gammaLogMLE *GammaModel::gammaLogMLE*

Description

Finds the Maximum Likelihood Estimate for a Gamma distribution, given some age-class count data

Usage

```
gammaLogMLE(counts, class.ages = NULL, trace = FALSE)
```

Arguments

`counts`	A one-row data frame comprising integer counts, with column names in capital letters. Formatting requirements detailed in `checkCountsFormat`.
`class.ages`	An optional one-row data frame specifying the starting age of each age-class. Age-classes as column names.
`trace`	An optional logical value indicating if the search progress should be printed.

Details

class.ages must be a one-row data frame giving the starting age of each age class or age multi-class. These must be sequential, such that the start of the (n+1)th class equals the end of the nth class. Since all possible ages must be accounted for, the first starting age must be zero, whilst the final age class will automatically include ages above this final age.

If the default class.ages = NULL is used, Payne's caprine age classes are used as follows, where units are in years:

class.ages <- data.frame(A=0, B=1/6, C=1/2, D=1, E=2, F=3, G=4, H=6, I=8)

The search for the Maximum Likelihood parameters is acheived using a bespoke jDE variant of the Differential Evolution stochastic algorithm for global optimization, as implemented by `JDEoptim` in Eduardo L T Conceicao's DEoptimR package.

Value

Single value numeric giving the exact log likelihood')

References

Kill-off Patterns in Sheep and Goats: The Mandibles from Asvan Kale. Payne 1973.

See Also

`JDEoptim`

Examples

```
# Maximum Likelihood Estimate of Gamma distribution
data(Neolithic)
counts <- Neolithic['TRA1',]
gamma.ll <- gammaLogMLE(counts)
```

gammaMLparameters *GammaModel::gammaMLparameters*

Description

Finds the Maximum Likelihood Gamma parameters given some age-class count data

Usage

```
gammaMLparameters(counts, class.ages = NULL, trace = FALSE)
```

Arguments

counts	A one-row data frame comprising integer counts, with column names in capital letters. Formatting requirements detailed in checkCountsFormat.
class.ages	An optional one-row data frame specifying the starting age of each age-class. Age-classes as column names.
trace	An optional logical value indicating if the search progress should be printed.

Details

class.ages must be a one-row data frame giving the starting age of each age class or age multi-class. These must be sequential, such that the start of the (n+1)th class equals the end of the nth class. Since all possible ages must be accounted for, the first starting age must be zero, whilst the final age class will automatically include ages above this final age.

If the default class.ages = NULL is used, Payne's caprine age classes are used as follows, where units are in years:

class.ages <- data.frame(A=0, B=1/6, C=1/2, D=1, E=2, F= 3, G=4, H=6, I=8)

The search for the Maximum Likelihood parameters is acheived using a bespoke jDE variant of the Differential Evolution stochastic algorithm for global optimization, as implemented by JDEoptim in Eduardo L T Conceicao's DEoptimR package.

Value

Data frame giving the estimated Gamma parameters ('shape' and 'mean')

References

Kill-off Patterns in Sheep and Goats: The Mandibles from Asvan Kale. Payne 1973.

See Also

JDEoptim

Examples

```
# Maximum Likelihood Gamma Parameters of the 'TRA1' data
# using the default Payne caprine age classes.
data(Neolithic)
counts <- Neolithic['TRA1',]
par <- gammaMLparameters(counts)

# Plot the Gamma distribution defined by these parameters
# Note, dgamma() requires the shape and rate

x <- seq(0,10,length.out=1000)
y <- dgamma(x, shape = par$shape, rate = par$shape/par$mean )
plot(x, y, type='l', xlab='age', ylab='PD', main='Gamma')
```

GammaModel *GammaModel*

Description

Tools to model count data in discrete classes using the Gamma distribution

Details

Package to analyse age-at-death count data across discrete age classes.

Inferential tools where there are large uncertainties from a) small sample sizes, and b) multi-class assignments. Therefore typically applicable to Archaeological data.

Calculates the exact probability of observed data given either a Gamma model or an age-class model. Therefore unsuitable for data with large sample sizes in multi- class assignments, since all possible combinations are computed.

MLE (under either model) found using DEoptimR search algorithm.

Likelihood distribution (under Gamma model) found using MCMC Metropolis-Hastings algorithm.

Performs Goodness of Fit tests, and model comparison using AIC and BIC.

References

'Modelling caprine age-at-death profiles using the Gamma distribution' by Adrian Timpson, Rosalind E. Gillis, Katie Manning and Mark G. Thomas, published in Journal of Archaeological Science 2018

GOF *GammaModel::GOF*

Description

Estimates Goodness of Fit of the counts to the age-class model, using the Chi-squared test.

Usage

```
GOF(counts, model, N = 1e+04)
```

Arguments

counts	A one-row data frame comprising integer counts, with column names in capital letters. Formatting requirements detailed in checkCountsFormat.
model	A one-row data frame specifying the age-class model probabilities. Age-classes as column names.
N	Positive integer specifying the number of random arrangements to generate.

Details

The general approach to calculating a p-value for n observed counts given the model probabilities is to calculate the probability of each possible arrangement of the n counts, then sum the probabilities that are smaller or equal to the probability of the observed arrangement.

This approach can be easily achieved if each count was assigned to only one class, but becomes computationally expensive when dealing with the additional problem of multi-class assignments.

Instead, the p-value can be estimated by generating random arrangements of the observed data (this naturally takes care of the different frequencies of each arrangement), calculating a p-value for each using a chi- squared test, then calculating the mean p-value. With a large number of random arrangements this approaches the exact probability of the observed data being as or more extreme.

This should not be confused methods of combining p-values such as Fisher's, which solves the different problem of combining p-values from independent trials.

Value

P-value in scientific notation, with two significant figures.

References

Pearson, K., 1900. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 50, 157-175

Examples

```
# How well does the 'FON1' data fit Payne's meat model
data(Neolithic)
data(models.payne)
counts <- Neolithic['FON1',]
model <- models.payne['meat',]
```

```
GOF(counts, model)
```

mcmc *GammaModel::mcmc*

Description

Generates a single Markov Chain Monte Carlo (MCMC) chain of Gamma distribution parameter samples, using the Metropolis-Hastings algorithm.

Usage

```
mcmc(counts, class.ages = NULL, N = 30000, burn = 2000, thin = 5, prop = 0.4, plot.chain = TRUE)
```

Arguments

`counts`	A one-row data frame comprising integer counts, with column names in capital letters. Formatting requirements detailed in `checkCountsFormat`.
`class.ages`	An optional one-row data frame specifying the starting age of each age-class. Age-classes as column names.
`N`	An optional integer specifying how many samples in the chain.
`burn`	An optional integer specifying how many initial samples in the chain should be discarded for burn-in.
`thin`	An optional integer specifying the proportion of samples to discard. I.e. 5 = every 5th sample in the chain is retained.
`prop`	An optional numerical value controlling the average jump size in the proposal function.
`plot.chain`	An optional logical value indicating if the chain should be plotted after completion.

Details

Prints progress every 1000th sample, and the final acceptance ratio (AR). For an efficient chain this should be around 0.3 to 0.6. If the AR is too low, efficiency can be improved by lowering the value of the prop argument.

The chain can be inspected to ensure good mixing using the default plot.chain = TRUE.

Value

Data frame of two columns, giving the Gamma parameters ('shape' and 'mean') sampled from the chain, after burn-in and thinning.

References

Hastings, W.K., 1970. Monte Carlo sampling methods using Markov chains and their applications, Biometrika 57, 97-109.

Examples

```
# Sampling from the joint likelihood distribution for TRA1
# using the default Payne class ages.
data(Neolithic)
counts <- Neolithic['TRA1',]
pars <- mcmc(counts)

# plot the final samples
par(mfrow = c(1, 1)  )
plot(pars,  xlab = 'shape',  ylab = 'mean' )
```

models.payne *GammaModel::models.payne*

Description

Probabilities of death in Payne's age classes A to I, proposed by Payne to represent the ideal kill strategies when attempting to maximise the meat, milk or wool yields from a caprine herd.

Usage

```
Neolithic
```

Format

A data frame comprising 27 probabilities, across 3 rows and 9 columns.

Source

Data as numeric probabilities was first published by Timpson et al 2018. However, this numeric summary was derived from the original source data resented graphically as survivorship curves in Table 1 of Payne, S., 1973.

References

Modelling caprine age-at-death profiles using the Gamma distribution, Adrian Timpson, Rosalind E. Gillis, Katie Manning and Mark G. Thomas, Journal of Archaeological Science 2018

Payne, S., 1973. Kill-off patterns in sheep and goats: the mandibles from Asvan Kale, Anatolian studies.

`models.redding` *GammaModel::models.redding*

Description

Probabilities of death in Payne's age classes A to I, proposed by Redding to represent the ideal kill strategies when attempting to maximise energy yields or herd security from a caprine herd.

Usage

```
Neolithic
```

Format

A data frame comprising 16 probabilities, across 2 rows and 8 columns.

Source

Data as numeric probabilities was first published in Redding, R.,1981.

References

Redding, R.,1981. Decision making in subsistence herding of sheep and goats in the Middle East. University of Michigan, Ann Arbor. Ph. D.

`Neolithic` *GammaModel::Neolithic*

Description

Age-at-death data comprising counts of caprine teeth from 10 European Neolithic sites, classified into Payne's age classes A to I. Some individual teeth have been assigned to several possible age classes (a multi-class).

Usage

```
Neolithic
```

Format

A data frame comprising 719 counts, across 10 rows and 18 columns

Source

Original source data first published in Timpson et al 2018.

References

Modelling caprine age-at-death profiles using the Gamma distribution. Adrian Timpson, Rosalind E. Gillis, Katie Manning and Mark G. Thomas. Journal of Archaeological Science 2018

Index