Package ‘SPEC’

July 8, 2021

Title Supervised Classification under Partition Exchangeability

Version 0.0.0.9000

Description This package implements a supervised predictive classifier underpartition echangeability due to J.F.C. Kingman (1978). The two classifiers implemented are a Naive Bayes classifier that assumes test datais i.i.d., and a more computationally costly but more accurate simultaneous classifier that tries to find a labeling for the entire test dataset at once. Also included in this package are functions to simulate samples and gain sample probabilities from the Ewens sampling formula, also known as the Poisson-Dirichlet distribution. Finally, parameter estimation and a simple hypothesis test for the distribution are included.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1.9001

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Imports gmp (>= 0.6.2),
stats (>= 4.1.0),
Rmpfr (>= 0.8.4)

R topics documented:

`abundances` *Vector of frequencies of frequencies*

Description

A function to calculate frequencies of frequencies or the combined frequencies of frequencies of two vectors.

Usage

```
abundances(freqs, freqs0 = NULL)
```

Arguments

`freqs`	A frequency table of data vector `x`.
`freqs0`	A second optional frequency table to be merged to the `freqs` parameter.

Details

`Freqs0` is optional. It is used in the case of calculating abundances of test data when the frequencies of the training data are already known. `Freqs0` is `table(x0)` ,where `x0` is training data. Abundances of any kind of data vector `x` can be calculated with `table(table(x))`.

Value

This function returns a named vector that is used to calculate probabilities and make classifications.

Examples

```
set.seed(111)
x<-rpois(10,10)
abundances(table(x))

y<-rpois(2,10)
abundances(table(x), table(y))
```

`dPD` *The Poisson-Dirichlet distribution*

Description

Distribution function for the Poisson-Dirichlet distribution.

Usage

```
dPD(abund, psi)
```

Arguments

abund	An abundance vector.
psi	Dispersal parameter. Accepted values are positive numbers, "a" for absolute value ψ=1 by default, or "r" for relative value psi equals sample size.

Details

dPD calculates the probability of a data vector x given by the Poisson-Dirichlet distribution, also known as the Ewens sampling formula. The higher the dispersal parameter ψ, the higher the amount of distinct observed species. In terms of the paintbox process, a high ψ increases the size of the continuous part p_0 of the process, while a low ψ will increase the size of the discrete parts $p_>0$.

Value

dPD returns the probability of the abundance vector of the data vector x, given dispersal parameter ψ.

References

W.J. Ewens, The sampling theory of selectively neutral alleles, Theoretical Population Biology, Volume 3, Issue 1, 1972, Pages 87-112, ISSN 0040-5809, https://doi.org/10.1016/0040-5809(72)90035-4.

Examples

```
## Get a random sample from the Poisson Dirichlet distribution, and
## find the probability of such a sample with psi=5:
set.seed(111)
s <- rPD(n=100,psi=5)
a=table(table(s))
dPD(a, psi=5)
```

LMTp *Lagrange Multiplier Test for psi*

Description

Performs the Lagrange Multiplier test for an abundance vector under partition exchangeability. Returns a p-value for the hypothesis that the input data vector stems from a population with the input dispersal parameter.

Usage

```
LMTp(abund, psi = "a")
```

Arguments

abund	An abundance vector.
psi	Target ψ to be tested. psi = "a" for absolute value 1, "r" for relative value n (sample size), or any positive number.

Details

$$U(\psi_0)^2/I(\psi_0)$$

, where U is the log-likelihood function of ψ and I is its Fisher information. The statistic follows χ^2-distribution with 1 degree of freedom when the null hypothesis $H_0 : \psi = \psi_0$ is true.

Value

A p-value.

References

Radhakrishna Rao, C, (1948), Large sample tests of statistical hypotheses concerning several parameters with applications to problems of estimation. Mathematical Proceedings of the Cambridge Philosophical Society, 44(1), 50-57. https://doi.org/10.1017/S0305004100023987

Examples

```
## Test the psi of a sample fro mthe Poisson-Dirichlet distribution:
set.seed(10000)
x<-rPD(1000, 10)
## Find the abundances of the data vector:
abund=abundances(table(x))
## Test for the psi that was used, as well as a high one and a low one:
LMTp(abund, 10)
LMTp(abund, 15)
LMTp(abund, 5)
LMTp(abund)       #test for psi=1
LMTp(abund, "r")  #test for psi=n
```

MLEp   *Maximum Likelihood Estimate for psi*

Description

Numerically searches for the MLE of psi given the frequencies of the frequencies of a data vector, called an abundance vector.

Usage

```
MLEp(abund)
```

Arguments

abund	An abundance vector.

Details

Numerically searches for the MLE of psi as the root of equation

$$K = \sum_{i=1}^{n} \psi/(\psi + i - 1),$$

where K is the observed number of different species in the sample. The right side of the equation is strictly increasing when $\psi > 0$, so a binary search is used to find the root. An accepted ψ sets value of the right side of the equation within R's smallest possible value of the actual value of K.

Value

Returns a list that contains the estimate "psi" and "Asymptotic confidence interval". The confidence interval is based on the asymptotic distribution of maximum likelihood estimators.

References

W.J. Ewens, The sampling theory of selectively neutral alleles, Theoretical Population Biology, Volume 3, Issue 1, 1972, Pages 87-112, ISSN 0040-5809, https://doi.org/10.1016/0040-5809(72)90035-4.

Examples

```
##Find the MLE of psi of the vector (1,2,2).
##The frequencies of the frequencies of the data vector are given as input:
MLEp(table(table(c(1,2,2))))

##Find the MLE of psi of a sample from the Poisson-Dirichlet distribution:
set.seed(1000)
x<-rPD(n=10000, psi=100)
MLEp(table(table(x)))
```

MLEp.bsci *Bootstrap confidence interval for the MLE of psi*

Description

A bootstrapped confidence interval for the Maximum Likelihood Estimate for psi based on a 100 bootstrap rounds with 80% of data used at a time.

Usage

```
MLEp.bsci(x, level)
```

Arguments

`x`	A data vector.
`level`	Level of confidence interval as number between 0 and 1.

Value

The MLE of psi as well as lower and upper bounds of the bootstrap confidence interval.

Examples

```
## Find a 95% -confidence interval for the MLE of psi given a sample from the
## Poisson-Dirichlet distribution:
x<-rPD(n=10000, psi=100)
MLEp.bsci(x, 0.95)
```

`rPD` *Random sampling from the Poisson-Dirichlet Distribution*

Description

rPD samples randomly from the PD distribution with a given ψ by simulating the Hoppe urn model.

Usage

```
rPD(n, psi)
```

Arguments

`n`	number of observations.
`psi`	dispersal parameter.

Details

`rPD` samples random values with a given ψ from the Poisson-Dirichlet distribution by simulating the Hoppe urn model.

Value

`rPD` returns a vector with a sample of size n from the Hoppe urn model with parameter ψ, along with its table of frequencies. given parameter ψ.

References

Hoppe, F.M. The sampling theory of neutral alleles and an urn model in population genetics. J. Math. Biology 25, 123–159 (1987). https://doi.org/10.1007/BF00276386.

W.J. Ewens, The sampling theory of selectively neutral alleles, Theoretical Population Biology, Volume 3, Issue 1, 1972, Pages 87-112, ISSN 0040-5809, https://doi.org/10.1016/0040-5809(72)90035-4.

Examples

```
## Get random sample from the PD distribution with different psi,
## and estimate the psi of the samples:
s1<-rPD(1000, 10)
s2<- rPD(1000, 50)
print(c(MLEp(table(table(s1)))$psi, MLEp(table(table(s2)))$psi))
```

`SPEC.fit` *Fit the supervised classifier under partition exchangeability*

Description

Trains the model according to training data x, where x is assumed to follow the Poisson-Dirichlet distribution, and discrete labels y.

Usage

```
SPEC.fit(x, y)
```

Arguments

`x`	data vector, or matrix with rows as data points and columns as features.
`y`	training data label vector of length equal to the amount of rows in `x`.

Value

Returns an object used as training data objects for the classification algorithms `tMarLab` and `tSimLab`.

If `x` is multidimensional, each list described below is returned for each dimension.

Returns a list of classwise lists, each with components:

`frequencies`: the frequencies of values in the class.

`psi`: the Maximum Likelihood estimate for ψ for the class.

Examples

```
## Create training data x and its class labels y from Poisson-Dirichlet distributions
## with different psis:
set.seed(111)
x1<-rPD(5000,10)
x2<-rPD(5000,100)
x<-c(x1,x2)
y1<-rep("1", 5000)
y2<-rep("2", 5000)
y<-c(y1,y2)
fit<-SPEC.fit(x,y)

## With multidimensional x:
```

```
set.seed(111)
x1<-cbind(rPD(5000,10),rPD(5000,50))
x2<-cbind(rPD(5000,100),rPD(5000,500))
x<-rbind(x1,x2)
y1<-rep("1", 5000)
y2<-rep("2", 5000)
y<-c(y1,y2)
fit<-SPEC.fit(x,y)
```

tMarLab *Marginally predicted labels of the test data given training data classification.*

Description

tMarLab classifies the test data x based on the training data object. The test data is considered i.i.d. and to have arrived sequentially. Thus, each data point is classified one by one.

Usage

```
tMarLab(training, x)
```

Arguments

training	A training data object from the function SPEC.fit.
x	Test data vector or matrix with rows as data points and columns as features.

Value

A vector of predicted labels for test data x.

References

Amiryousefi A. Asymptotic supervised predictive classifiers under partition exchangeability. . 2021. https://arxiv.org/abs/2101.10950.

Corander, J., Cui, Y., Koski, T., and Siren, J.: Have I seen you before? Principles of Bayesian predictive classification revisited. Springer, Stat. Comput. 23, (2011), 59–73. (https://doi.org/10.1007/s11222-011-9291-7)

Examples

```
## Create random samples x from Poisson-Dirichlet distributions with different
## psis, treating each sample as coming from a class of its own:
set.seed(111)
x1<-rPD(10500,10)
x2<-rPD(10500,1000)
test.ind1<-sample.int(10500,500) # Sample test datasets from the
test.ind2<-sample.int(10500,500) # original samples
```

```
x<-c(x1[-test.ind1],x2[-test.ind2])
## create training data labels:
y1<-rep("1", 10000)
y2<-rep("2", 10000)
y<-c(y1,y2)

## Test data t, with first half belonging to class "1", second have in "2":
t1<-x1[test.ind1]
t2<-x2[test.ind2]
t<-c(t1,t2)

fit<-SPEC.fit(x,y)

## Run the classifier, which returns
tM<-tMarLab(fit, t)

##With multidimensional x:
set.seed(111)
x1<-cbind(rPD(5500,10),rPD(5500,50))
x2<-cbind(rPD(5500,100),rPD(5500,500))
test.ind1<-sample.int(5500,500)
test.ind2<-sample.int(5500,500)
x<-rbind(x1[-test.ind1,],x2[-test.ind2,])
y1<-rep("1", 5000)
y2<-rep("2", 5000)
y<-c(y1,y2)
fit<-SPEC.fit(x,y)
t1<-x1[test.ind1,]
t2<-x2[test.ind2,]
t<-rbind(t1,t2)

tM<-tMarLab(fit, t)
```

tSimLab *Simultaneously predicted labels of the test data given the training data classification.*

Description

tSimLab classifies the test data x based on the training data object. All of the test data is used simultaneously to make the classification.

Usage

```
tSimLab(training, x)
```

Arguments

training A training data object from the function SPEC.fit.

x Test data vector or matrix with rows as data points and columns as features.

Details

The simultaneous case: The test data are first labeled with the marginal classifier. The simultaneous classifier then iterates over all test data, assigning each a label by finding the maximum predictive probability given the current classification structure of the test data as a whole. This is repeated until the classification structure doesn't change after iterating over all data.

Value

A vector of predicted labels for test data x.

References

Amiryousefi A. Asymptotic supervised predictive classifiers under partition exchangeability. . 2021. https://arxiv.org/abs/2101.10950.

Corander, J., Cui, Y., Koski, T., and Siren, J.: Have I seen you before? Principles of Bayesian predictive classification revisited. Springer, Stat. Comput. 23, (2011), 59–73.(https://doi.org/10.1007/s11222-011-9291-7)

Examples

```
## Create random samples x from Poisson-Dirichlet distributions with different
## psis, treating each sample as coming from a class of its own:
set.seed(111)
x1<-rPD(10500,10)
x2<-rPD(10500,1000)
test.ind1<-sample.int(10500,500) # Sample test datasets from the
test.ind2<-sample.int(10500,500) # original samples
x<-c(x1[-test.ind1],x2[-test.ind2])
## create training data labels:
y1<-rep("1", 10000)
y2<-rep("2", 10000)
y<-c(y1,y2)

## Test data t, with first half belonging to class "1", second have in "2":
t1<-x1[test.ind1]
t2<-x2[test.ind2]
t<-c(t1,t2)

fit<-SPEC.fit(x,y)

## Run the classifier, which returns
tS<-tSimLab(fit, t)

##With multidimensional x:
set.seed(111)
x1<-cbind(rPD(5500,10),rPD(5500,50))
x2<-cbind(rPD(5500,100),rPD(5500,500))
test.ind1<-sample.int(5500,500)
test.ind2<-sample.int(5500,500)
x<-rbind(x1[-test.ind1,],x2[-test.ind2,])
y1<-rep("1", 5000)
```

```
y2<-rep("2", 5000)
y<-c(y1,y2)
fit<-SPEC.fit(x,y)
t1<-x1[test.ind1,]
t2<-x2[test.ind2,]
t<-rbind(t1,t2)

tS<-tSimLab(fit, t)
```

WattersonsW *Watterson's homozygosity test*

Description

This function performs a statistical test on the null hypothesis thata given sample's underlying distribution is the Poisson-Dirichlet distribution. It calculates a test statistic that is then used to gain a p-value from an empirical distribution of the statistic from simulated samples from a PD distribution.

Usage

```
WattersonsW(x, rounds)
```

Arguments

`x`	A data vector.
`rounds`	How many samples are simulated to obtai nthe empirical distribution.

Value

A p-value.

References

Watterson, G.A., (1978), The homozygosity test of neutrality. Genetics. 88(2):405-417.

Examples

```
##Test whether a typical sample follows PD:
x<-rPD(1000,10)
WattersonsW(x, 1000)

##Test whether a very atypical sample where frequencies of different values
## are similar:

x<-c(rep(1, 200), rep(2, 200), rep(3, 200), rep(4, 200), rep(5,200))
WattersonsW(x,1000)
```

Index