Package 'SPEC'

June 29, 2021

Title Supervised Classification under Partition Exchangeability

Version 0.0.0.9000

Description This package implements a supervised predictive classifier under partition echangeability due to J.F.C. Kingman (1978). Given training data and labels, it learns the maximum likelihood estimate for the single parameter of the so called Ewens sampling formula. The estimate along with the frequencies of feature values is then used to calculate predictive probabilities for test data. The two classifiers implemented are a Naive Bayes classifier that assumes test data is 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 stats

R topics documented:

---

abundances *Abundances - frequencies of frequencies*

Description

A function to calculate frequencies of frequencies or the combined frequencies of frequencies of two vectors. Freqs are calculated by table(x), where x is data. 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)). This returns a named vector that is used to calculate probabilities and make classifications.

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.

Examples

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

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

---

dlPD *Poisson Dirichlet distribution*

Description

LogProbability of a data vector x from Ewens' sampling formula is given below. accepts either a raw data vector x or its frequency vector, table(x). The higher the dispersal parameter the psi is, the higher the amount of distinct observed species will be. In terms of the paintbox process, a high psi increases the size of the continuous part p_0 of the process, while a low psi will increase the size of the discrete parts p_1, ... p_k.

Usage

```
dlPD(abund, psi)
```

Arguments

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

Value

dlPD returns the log-probability of the abundance vector of the data vector x,

Examples

```
set.seed(111)
s <- rPD(100,5)
a=abundances(table(s))
dlPD(a, 5)
```

---

dPD *Poisson Dirichlet distribution*

Description

Probability of a data vector x given by Ewens' sampling formula. The higher the dispersal parameter the psi is, the higher the amount of distinct observed species will be. In terms of the paintbox process, a high psi increases the size of the continuous part p_0 of the process, while a low psi will increase the size of the discrete parts p_>0.

Usage

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

Arguments

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

Value

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

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

```
set.seed(111)
s <- rPD(100,5)
a=table(table(s))
dPD(a, 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 psi 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 psi and I is its Fisher information. The statistic follows chi-squared distribution with 1 degree of freedom when the null hypothesis H_0:psi<-psi_0 is true.

Value

A p-value.

Examples

```
set.seed(10000)
x<-rPD(1000, 10)
abund=abundances(table(x))
LMTp(abund, 10)
LMTp(abund, 15)
LMTp(abund, 5)
LMTp(abund)       #test for psi=1
LMTp(abund, "r")  #test for psi=n
```

lognRF *Log of rising factorial psi(psi+1)...(psi+n-1)*

Description

lognRF calculates log(psi)+log(psi+1)+...+log(psi+n-1).

Usage

```
lognRF(psi, n)
```

MLEp *Maximum Likelihood Estimate for psi*

Description

Numerically searches for the MLE of psi as the root of equation K=sum(psi/psi+i-1) for i<-1:n, where K is the observed number of different species in the sample. An accepted psi sets the value of the right side of the equation within R's smallest possible value of the actual value of K. 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.

Usage

```
MLEp(abund)
```

Arguments

abund	An abundance vector.

Details

Numerically searches for the MLE of psi as the root of equation K<-sum(psi/psi+i-1) for i<-1:n, where K is the observed number of different species in the sample. An accepted psi 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 containing the MLE in $psi and an asymptotic confidence interval in $´Asymptotic confidence interval´

Examples

```
MLEp(abundances(table(c(1,2,2))))

set.seed(1000)
x<-rPD(10000,2000)
MLEp(abundances(table(x)))
```

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

Description

A bootstrapped confidence interval for the Maximum Likelihood Estimate for psi.

Usage

```
MLEp.bsci(x)
```

Arguments

x	A data vector.

Value

Lower and upper bounds of the confidence interval.

rPD *Sampling from he Dirichlet-Poisson Distribution*

Description

rPD samples from the PD distribution by simulating the Hoppe urn model.

Usage

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

Arguments

n	number of observations.
psi	dispersal parameter.

Value

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

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

Examples

```
set.seed(111)
s <- rPD(100,5)
```

SPEC.fit *Fit the supervised classifier*

Description

Trains the model according to training data x and labels y. The output is a classwise list including the frequencies of the data, and the MLE of psi.

Usage

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

Arguments

x	data vector, or matrix with rows as data points and columns as features.
y	training label vector.

Value

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 estimate of psi for the class.

Examples

```
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

#' The classification algorithm is adapted from 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

```
set.seed(111)
x1<-rPD(10500,10)
x2<-rPD(10500,1000)
test.ind1<-sample.int(10500,500)
test.ind2<-sample.int(10500,500)
x<-c(x1[-test.ind1],x2[-test.ind2])
y1<-rep("1", 10000)
y2<-rep("2", 10000)
y<-c(y1,y2)

t1<-x1[test.ind1]
t2<-x2[test.ind2]
t<-c(t1,t2)

fit<-SPEC.fit(x,y)

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

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. The classification algorithm is adapted from 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)

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.

Value

A vector of predicted labels for test data x.

References

The classification algorithm is adapted from 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

```
set.seed(111)
x1<-rPD(11500,10)
x2<-rPD(11500,1000)
test.ind1<-sample.int(10500,500)
test.ind2<-sample.int(10500,500)
x<-c(x1[-test.ind1],x2[-test.ind2])
y1<-rep("1", 10000)
y2<-rep("2", 10000)
y<-c(y1,y2)

t1<-x1[test.ind1]
t2<-x2[test.ind2]
t<-c(t1,t2)

fit<-SPEC.fit(x,y)

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)
```

Index