Package: xactonomial (via r-universe)

October 16, 2024

Type Package

Title Exact Inference for Real-Valued Functionals of k-Sample Multinomial Parameters

Version 0.4.0

Date 2024-06-18

URL https://github.com/sachsmc/xactonomial

BugReports https://github.com/sachsmc/xactonomial/issues

Description We consider the k sample multinomial problem where we observe k vectors (possibly of different lengths), each representing an independent sample from a multinomial. For a given function psi which takes in the concatenated vector of multinomial probabilities and outputs a real number, we are interested in constructing a confidence interval for psi. We use an exact (but computational and stochastic) method to compute a confidence interval and a function for calculation of p values. The details of the method will be in a forthcoming preprint.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Suggests knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.3.1

Config/rextendr/version 0.3.1.9000

Repository https://sachsmc.r-universe.dev

RemoteUrl https://github.com/sachsmc/xactonomial

RemoteRef HEAD

RemoteSha 7c0820c1c98edbdeab8ad05952ad21b987c524d1

Contents

`calc_probs_rust` *calculate multinomial probabilities*

Description

calculate multinomial probabilities

Usage

```
calc_probs_rust(sar, logt, logc, d, n, nt)
```

`calc_prob_null` *Calculate probability for given parameters*

Description

Given a set of candidate parameter vectors, check if the null $\psi \leq \psi_0$ is satisfied, and if so, compute the probability for each element of the sample space

Usage

```
calc_prob_null(theta_cands, psi, psi0, minus1, SSpacearr, logC, II)
```

Arguments

theta_cands	A matrix with samples in the rows and the parameters in the columns
psi	The function of interest mapping parameters to the real line
psi0	The null boundary for testing psi <= psi0
minus1	Either plus or minus 1
SSpacearr	A matrix with the sample space for the given size of the problem
logC	log multinomial coefficient for each element of the sample space
II	logical vector of sample space psi being more extreme than the observed psi

Value

A numeric vector of probabilities

---

calc_prob_null2 *Calculate probability for given parameters*

Description

Given a set of candidate parameter vectors, check if the null $\psi \leq \psi_0$ is satisfied, and if so, compute the probability for each element of the sample space

Usage

```
calc_prob_null2(theta_cands, psi, psi0, minus1, SSpacearr, logC, II)
```

Arguments

theta_cands	A matrix with samples in the rows and the parameters in the columns
psi	The function of interest mapping parameters to the real line
psi0	The null boundary for testing psi <= psi0
minus1	Either plus or minus 1
SSpacearr	A matrix with the sample space for the given size of the problem
logC	log multinomial coefficient for each element of the sample space
II	logical vector of sample space psi being more extreme than the observed psi

Value

A numeric vector of probabilities

expand_index — *Get a matrix of indices for all possible combinations of vectors of lengths*

Description

This is basically the same as expand.grid, but faster for integers

Usage

```
expand_index(lengths)
```

Arguments

`lengths`	A vector with the lengths of each index to expand

Value

A matrix with length(lengths) columns and prod(lengths) rows

Examples

```
expand_index(c(2, 3, 4))
## the same as
expand.grid(1:2, 1:3, 1:4)
```

get_theta_random — *Sample from the unit simplex in d dimensions*

Description

Sample from the unit simplex in d dimensions

Usage

```
get_theta_random(d = 4, nsamp = 75)
```

Arguments

`d`	the dimension
`nsamp`	the number of samples to take uniformly in the d space

Value

The grid over Theta, the parameter space. A matrix with d columns and nsamp rows

Examples

```
get_theta_random(3, 10)
```

---

itp_root *Find the root of the function f*

Description

This finds the value $x \in [a, b]$ such that $f(x) = 0$ using the one-dimensional root finding ITP method (Interpolate Truncate Project). Also see itp.

Usage

```
itp_root(
  f,
  a,
  b,
  k1 = 0.1,
  k2 = 2,
  n0 = 1,
  eps = 0.005,
  maxiter = 100,
  fa = NULL,
  fb = NULL,
  verbose = FALSE,
  ...
)
```

Arguments

f	The function to find the root of in terms of its first (one-dimensional) argument
a	The lower limit
b	The upper limit
k1	A tuning parameter
k2	Another tuning parameter
n0	Another tuning parameter
eps	Convergence tolerance
maxiter	Maximum number of iterations
fa	The value of f(a), if NULL then will be calculated
fb	The value of f(b), if NULL then will be calculated
verbose	Prints out information during iteration
...	Other arguments passed on to f

Value

A numeric vector of length 1, the root at the last iteration

References

I. F. D. Oliveira and R. H. C. Takahashi. 2020. An Enhancement of the Bisection Method Average Performance Preserving Minmax Optimality. ACM Trans. Math. Softw. 47, 1, Article 5 (March 2021), 24 pages. https://doi.org/10.1145/3423597

Examples

```
fpoly <- function(x) x^3 - x - 2 ## example from the ITP_method wikipedia entry
itp_root(fpoly, 1, 2, eps = .0001, verbose = TRUE)
```

---

log_multinom_coef   *Calculate log of multinomial coefficient*

Description

Calculate log of multinomial coefficient

Usage

```
log_multinom_coef(x, sumx)
```

Arguments

x	Vector of observed counts in each cell
size	Total count

Value

The log multinomial coefficient

Examples

```
#' @examples
S0 <- sspace_multinom(4, 6)
S1 <- sspace_multinom(4, 7)
logC0<- apply(S0,1,log_multinom_coef,sumx=6)
logC1<- apply(S1,1,log_multinom_coef,sumx=7)
logC<- outer(logC0,logC1,'+')
```

`pvalue_psi0` *Compute a p value for the test of psi <= psi0*

Description

Compute a p value for the test of psi <= psi0

Usage

```
pvalue_psi0(
  psi0,
  psi,
  psi_hat,
  psi_obs,
  maxit,
  chunksize,
  lower = TRUE,
  target,
  SSpacearr,
  logC,
  d_k
)
```

Arguments

`psi0`	The null value
`psi`	The function of interest
`psi_hat`	The vector of psi values at each element of the sample space
`psi_obs`	The observed estimate
`maxit`	Maximum iterations
`chunksize`	Chunk size
`lower`	Do a one sided test of the null that it is less than psi0, otherwise greater.
`target`	Stop the algorithm if p >= target (for speed)
`SSpacearr`	The sample space array
`logC`	The log multinomial coefficient
`d_k`	The vector of dimensions

Value

A p-value

`rust_sspace` *calculate sample space of a multinomial with dimension d and sample size n*

Description

calculate sample space of a multinomial with dimension d and sample size n

Usage

```
rust_sspace(d, n)
```

`sample_unit_simplex` *Return a random sample from the d unit simplex*

Description

Return a random sample from the d unit simplex

Usage

```
sample_unit_simplex(d)
```

`sample_unit_simplexn` *Return n random samples from the d unit simplex*

Description

Return n random samples from the d unit simplex

Usage

```
sample_unit_simplexn(d, n)
```

sspace_multinom — *Enumerate the sample space of a multinomial*

Description

We have d mutually exclusive outcomes and n independent trials. This function enumerates all possible vectors of length d of counts of each outcome for n trials, i.e., the sample space. The result is output as a matrix with d columns where each row represents a possible observation.

Usage

```
sspace_multinom(d, n)
```

Arguments

`d`	Dimension
`n`	Size

Value

A matrix with d columns

Examples

```
d4s <- sspace_multinom(4, 8)
stopifnot(abs(sum(apply(d4s, 1, dmultinom, prob = rep(.25, 4))) - 1) < 1e-12)
```

xactonomial — *Exact inference for a real-valued function of multinomial parameters*

Description

We consider the k sample multinomial problem where we observe k vectors (possibly of different lengths), each representing an independent sample from a multinomial. For a given function psi which takes in the concatenated vector of multinomial probabilities and outputs a real number, we are interested in constructing a confidence interval for psi.

Usage

```
xactonomial(
  psi,
  data,
  psi0 = NULL,
  alpha = 0.05,
  psi_limits,
  maxit = 50,
  chunksize = 500,
  conf.int = TRUE
)
```

Arguments

psi	Function that takes in a vector of parameters and outputs a real valued number
data	A list with k elements representing the vectors of counts of a k-sample multinomial
psi0	The null hypothesis value. A p value for a test of psi <= psi0 is computed. If NULL only a confidence interval is computed.
alpha	A 1 - alpha percent confidence interval will be computed
psi_limits	A vector of length 2 giving the lower and upper limits of the range of $\psi(\theta)$
maxit	Maximum number of iterations of the stochastic procedure
chunksize	The number of samples taken from the parameter space at each iteration
conf.int	Logical. If FALSE, no confidence interval is calculed, only the p-value.

Details

Let T_j be distributed Multinomial$_{d_j}(\boldsymbol{\theta}_j, n_j)$ for $j = 1, \ldots, k$ and denote $\boldsymbol{T} = (T_1, \ldots, T_k)$ and $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)$. The subscript d_j denotes the dimension of the multinomial. Suppose one is interested in the parameter $\psi(\boldsymbol{\theta}) \in \Psi \subseteq \mathbb{R}$. Given a sample of size n from $\boldsymbol{T}$, one can estimate $\boldsymbol{\theta}$ with the sample proportions as $\hat{\boldsymbol{\theta}}$ and hence $\psi(\hat{\boldsymbol{\theta}})$. This function constructs a $1 - \alpha$ percent confidence interval for $\psi(\boldsymbol{\theta})$ and provides a function to calculate a p value for a test of the null hypothesis $H_0 : \psi(\boldsymbol{\theta}) \leq \psi_0$. We make no assumptions and do not rely on large sample approximations. The computation in somewhat involved so it is best for small sample sizes.

Value

A list with 3 elements: the estimate, the 1 - alpha percent confidence interval, and p-value

Examples

```
psi_ba <- function(theta) {
  theta1 <- theta[1:4]
  theta2 <- theta[5:8]
  sum(sqrt(theta1 * theta2))
  }
data <- list(T1 = c(2,1,2,1), T2 = c(0,1,3,3))
```

```
xactonomial(psi_ba, data, psi_limits = c(0, 1), maxit = 5, chunksize = 20)
```

Index