Package ‘prob.4.2.comp.stats’

July 25, 2021

Title Solution for Problem 4.2 in Computation Statistics by Givens

Version 1.0.0.0

Description Implements EM point estimator and its covariance estimator for problem 4.2 in Computation Statistics by Givens.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1.9001

R topics documented:

`em.counts_to_responses` *Convert counts to individual responses*

Description

There are many ways to represent the same data. In problem 4.2, the data is given in the form of counts. This is the inverse of em.counts_to_responses, e.g.,

```
xs <- em.counts_to_responses(counts)
em.responses_to_counts(xs) == counts
em.counts_to_responses(em.responses_to_counts(xs)) == xs.
```

Usage

```
em.counts_to_responses(counts)
```

Arguments

`counts`	count data

Value

the count data converted to response data, say counts = (379,299,...), then 379 responded 0 encounters, 299 responded 1 encounter, ...

Examples

```
# let counts be the count data
counts[j] # denotes number of respondents with j risky sexual encounters.
x <- em.counts_to_responses(counts)
x[k] # denotes the response of the i-th person
```

`em.cov.bs` *Bootstrap covariance estimator of EM point estimator*

Description

Bootstrap covariance estimator of EM point estimator

Usage

```
em.cov.bs(
  theta.em,
  counts,
  m = 10000,
  em.eps = 1e-06,
  bs.eps = 1e-04,
  debug = F
)
```

Arguments

`theta.em`	An EM point estimator of theta given observed counts
`counts`	observed count data (n0,n1,...,n16)
`m`	maximum bootstrap replicates
`em.eps`	EM algorithm epsilon stopping condition
`bs.eps`	Bootstrap covariance epsilon stopping condition
`debug`	whether to print out debugging info while running

`em.cov.info` *Covariance matrix of EM point estimator based on the observed information matrix*

Description

Covariance matrix of EM point estimator based on the observed information matrix

Usage

```
em.cov.info(theta.em, counts)
```

Arguments

`theta.em`	An EM point estimator of theta given observed responses
`data`	Observed sample of responses

`em.estimator` *EM algorithm*

Description

EM algorithm estimator for problem 4.2

Usage

```
em.estimator(theta, counts, eps = 1e-06, debug = T)
```

Arguments

`theta`	a starting guess for theta = (alpha,beta,mu,lambda)
`counts`	observed count data (n0,n1,...,n16)
`eps`	stopping condition
`debug`	whether to print out debugging info while running

`em.loglike` *log-likelihood function for problem 4.2*

Description

log-likelihood function for problem 4.2

Usage

```
em.loglike(theta, data)
```

Arguments

`theta`	evaluated at theta = (alpha,beta,mu,lambda)
`data`	observed response data

`em.observed_info` *Observed information of EM point estimator based on an observed sample*

Description

Observed information of EM point estimator based on an observed sample

Usage

```
em.observed_info(theta.em, data)
```

Arguments

`theta.em`	An EM point estimator of theta given observed counts
`data`	Observed sample of responses

em.pdf *probability density function (pdf) for problem 4.2*

Description

probability density function (pdf) for problem 4.2

Usage

```
em.pdf(x, theta)
```

Arguments

`x`	density evaluated at x given index theta
`theta`	index of indexed family of density functions, theta = (alpha,beta,mu,lambda)

em.responses_to_counts *Convert individual response data to count data*

Description

This is the inverse of em.counts_to_responses, e.g., em.responses_to_counts(em.counts_to_responses(counts)) == counts and em.counts_to_responses(em.responses_to_counts(data)) == data.

Usage

```
em.responses_to_counts(data)
```

Arguments

`data`	response data

Value

response data converted to count data

Index