Package ‘EBCHS’

February 14, 2026

Type Package

Title An Empirical Bayes Method for Chi-Squared Data

Version 0.1.1

Author Lilun Du [aut, cre],
Inchi Hu [aut]

Maintainer Lilun Du <lilundu@cityu.edu.hk>

Description We provide the main R functions to compute the posterior interval for the noncentrality parameter of the chi-squared distribution. The skewness estimate of the posterior distribution is also available to improve the coverage rate of posterior intervals. Details can be found in Du and Hu (2022) <doi:10.1080/01621459.2020.1777137>.

License MIT + file LICENSE

Encoding UTF-8

URL https://github.com/dulilun/EBCHS

RoxygenNote 7.3.3

Imports stats, pracma, splines, fda

Suggests testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2026-02-14 14:50:02 UTC

Contents

density_g_model — *The l_1 to l_4 derivative from the g-modeling method*

Description

The l_1 to l_4 derivative from the g-modeling method

Usage

```
density_g_model(x, k, pi_0, lambda_set, g_prior)
```

Arguments

x	a sequence of chi-squared test statistics
k	degrees of freedom
pi_0	the proportion of the null
lambda_set	the set of noncentrality values
g_prior	the prior probability for the noncentrality values

Value

a list: the margianl density, and its first-to-fourth derivatives

density_LS — *log-density derivatives–parametric approach*

Description

Assuming the log density of the chi-squared statistics admits a parametric form, this function estimates up to the fourth order log-density derivatives.

Usage

```
density_LS(x)
```

Arguments

x	a sequence of chi-squared test statistics

Value

a list: the first-to-fourth log density derivatives

Examples

```
p = 1000
k = 7
# the prior distribution for lambda
alpha = 2
beta =  10
# lambda
lambda = rep(0, p)
pi_0 = 0.8
p_0 = floor(p*pi_0)
p_1 = p-p_0
lambda[(p_0+1):p] = stats::rgamma(p_1, shape = alpha, rate=1/beta)
# Generate a Poisson RV
J = sapply(1:p, function(x){rpois(1, lambda[x]/2)})
X = sapply(1:p, function(x){rchisq(1, k+2*J[x])})
out = density_LS(X)
```

---

density_PLS *Penalized least-squares method in Du and Hu (2022)*

Description

The semiparametric model is employed to estimate the log density derivatives of the chi-squared statistics.

Usage

```
density_PLS(x, qq)
```

Arguments

x	a sequence of chi-squared test statistics
qq	the quantiles used for splines

Value

a list: the first and second density derivatives

Examples

```
p = 1000
k = 7
# the prior distribution for lambda
alpha = 2
beta =  10
# lambda
lambda = rep(0, p)
pi_0 = 0.5
```

```
p_0 = floor(p*pi_0)
p_1 = p-p_0
lambda[(p_0+1):p] = stats::rgamma(p_1, shape = alpha, rate=1/beta)
# Generate a Poisson RV
J = sapply(1:p, function(x){rpois(1, lambda[x]/2)})
X = sapply(1:p, function(x){rchisq(1, k+2*J[x])})
qq = c(0.2, 0.4, 0.6, 0.8)
out = density_PLS(X, qq)
```

EB_CS *Main function used in the paper (Du and Hu, 2022)*

Description

Give a sequence of chi-squared statistic values, the function computes the posterior mean, variance, and skewness of the non-centrality parameter given the data.

Usage

```
EB_CS(
  x,
  df,
  qq = c(0.2, 0.4, 0.6, 0.8),
  method = c("LS", "PLS", "g_model"),
  mixture = FALSE
)
```

Arguments

x	a sequence of chi-squared test statistics
df	the degrees of freedom
qq	the quantiles used in spline basis
method	LS: parametric least-squares; PLS: penalized least-squares; g-model: g-modeling
mixture	default is FALSE: there is no point mass at zero.

Value

a list: posterior mean, variance, and skewness estimates

References

Du and Hu (2022), *An Empirical Bayes Method for Chi-Squared Data*, *Journal of American Statistical Association*, forthcoming.

Examples

```
p = 1000
k = 7
# the prior distribution for lambda
alpha = 2
beta =  10
# lambda
lambda = rep(0, p)
pi_0 = 0.8
p_0 = floor(p*pi_0)
p_1 = p-p_0
lambda[(p_0+1):p] = rgamma(p_1, shape = alpha, rate=1/beta)
# Generate a Poisson RV
J = sapply(1:p, function(x){rpois(1, lambda[x]/2)})
X = sapply(1:p, function(x){rchisq(1, k+2*J[x])})
qq_set = seq(0.01, 0.99, 0.01)
out = EB_CS(X, k, qq=qq_set, method='LS', mixture = TRUE)
E = out$E_lambda
V = out$V_lambda
S = out$S_lambda
```

---

predictive_recursion *Predictive recursion by Newton (2002)*

Description

Predictive recursion by Newton (2002)

Usage

```
predictive_recursion(x, k)
```

Arguments

x	a sequence of chi-squared test statistics
k	degrees of freedom

Value

a list: null proportion, prior probability, and lambda-mesh values

Examples

```
set.seed(2021)
p = 1000
k = 7
# the prior distribution for lambda
alpha = 2
```

```
beta =  10
# lambda
lambda = rep(0, p)
pi_0 = 0
p_0 = floor(p*pi_0)
p_1 = p-p_0
lambda[(p_0+1):p] = stats::rgamma(p_1, shape = alpha, rate=1/beta)
# Generate a Poisson RV
J = sapply(1:p, function(x){rpois(1, lambda[x]/2)})
X = sapply(1:p, function(x){rchisq(1, k+2*J[x])})
out = predictive_recursion(X, k)
```

Index