Package ‘OneStep’

July 13, 2020

Type Package

Title One-Step Maximum Likelihood Estimation

Version 0.9.0

Author Alexandre Brouste, Darel Noutsa Mieniedou, Christophe Dutang

Maintainer Christophe Dutang <christophe.dutang@ensimag.fr>

Contact Alexandre Brouste, Darel Noutsa Mieniedou, Christophe Dutang

<onestep@univ-lemans.fr>

Description Provide principally an eponymic function that numerically computes the Le Cam one-step estimator which is asymptotically efficient (see e.g. L. Le Cam (1956) <https://projecteuclid.org/euclid.bsmsp/1200501652>) and can be computed faster than the maximum likelihood estimator.

License GPL (>= 2)

Encoding UTF-8

Depends fitdistrplus, numDeriv

Suggests actuar

NeedsCompilation no

Repository CRAN

Date/Publication 2020-07-13 10:00:06 UTC

R topics documented:

OneStep-package	*One-Step Maximum Likelihood Estimation*

Description

Provide principally an eponymic function that numerically computes the Le Cam one-step estimator which is asymptotically efficient (see e.g. L. Le Cam (1956) <https://projecteuclid.org/euclid.bsmsp/1200501652>) and can be computed faster than the maximum likelihood estimator.

Details

The DESCRIPTION file:

Package:	OneStep
Type:	Package
Title:	One-Step Maximum Likelihood Estimation
Version:	0.9.0
Author:	Alexandre Brouste, Darel Noutsa Mieniedou, Christophe Dutang
Maintainer:	Christophe Dutang <christophe.dutang@ensimag.fr>
Contact:	Alexandre Brouste, Darel Noutsa Mieniedou, Christophe Dutang <onestep@univ-lemans.fr>
Description:	Provide principally an eponymic function that numerically computes the Le Cam one-step estimator which is a
License:	GPL (>= 2)
Encoding:	UTF-8
Depends:	fitdistrplus, numDeriv
Suggests:	actuar

Index of help topics:

```
OneStep-package         One-Step Maximum Likelihood Estimation
benchonestep            Performing benchmark of one-step MLE against
                        other methods
onestep                 Executing Le Cam's one-step estimation
```

one-step MLE estimation

Author(s)

Alexandre Brouste, Darel Noutsa Mieniedou, Christophe Dutang

Maintainer: Christophe Dutang <christophe.dutang@ensimag.fr>

References

L. LeCam (1956). *On the asymptotic theory of estimation and testing hypothesis*. In: Proceedings of 3rd Berkeley Symposium I, pages 355–368.

See Also

See `fitdistrplus` for classic MLE, MME,...

`benchonestep` *Performing benchmark of one-step MLE against other methods*

Description

Performing benchmark of one-step MLE against other methods

Usage

```
benchonestep(data, distr, methods, init, weights=NULL,...)
```

Arguments

`data`	A numeric vector of length `n`
`distr`	A character string `"name"` naming a distribution for which the corresponding density function dname and the corresponding distribution function pname must be classically defined.
`methods`	A list of methods.
`init`	A character vector for the intial guess method.
`weights`	UNUSED : an optional vector of weights to be used in the fitting process. Should be `NULL` or a numeric vector with strictly positive integers (typically the number of occurences of each observation). If non-`NULL`, weighted MLE is used, otherwise ordinary MLE.
`...`	unused

Value

An array.

Author(s)

Alexandre Brouste, Darel Noutsa Mieniedou, Christophe Dutang

References

L. LeCam (1956). *On the asymptotic theory of estimation and testing hypothesis*. In: Proceedings of 3rd Berkeley Symposium I, pages 355–368.

onestep *Executing Le Cam's one-step estimation*

Description

Executing Le Cam's one-step estimation based on Le Cam (1956) and Kamatani and Uchida (2015).

Usage

```
onestep(data, distr, method, init, weights = NULL,
  keepdata = TRUE, keepdata.nb=100, control=list(),  ...)
```

Arguments

`data`	A numeric vector of length `n`
`distr`	A character string `"name"` naming a distribution for which the corresponding density function dname and the corresponding distribution function pname must be classically defined.
`method`	A character string coding for the fitting method: `"closed formula"` for explicit one-step and `"numeric"` for numeric computation. The default method is the `"closed formula"`.
`init`	A character vector for the intial guess method.
`weights`	an optional vector of weights to compute the final likelihood. Should be `NULL` or a numeric vector with strictly positive integers (typically the number of occurences of each observation).
`keepdata`	a logical. If `TRUE`, dataset is returned, otherwise only a sample subset is returned.
`keepdata.nb`	When `keepdata=FALSE`, the length (>1) of the subset returned.
`control`	a list of control parameters. Currently, `param_t` is used when the characteristic function is needed, `delta` is used when the subsample of size `n^delta` is randomly selected for the initial guess in the generic Le Cam's one step method.
`...`	further arguments passe to `mledist` in case it is used.

Details

The Le Cam one-step estimation procedure is based on an initial sequence of guess estimators and a Fisher scoring step or a single Newton step on the loglikelihood function. For the user, the function onestep chooses automatically the best procedure to be used. The function OneStep presents internally several procedures depending on whether the sequence of initial guess estimators is in a closed form or not, and on whether the score and the Fisher information matrix can be elicited in a closed form. "Closed formula" distributions are treated with explicit score and Fisher information matrix (or Hessian matrix). For all other distributions, if the density function is well defined, the numerical computation (`NumDeriv`) of the Newton step in Le Cam's one-step is proposed with an initial sequence of guess estimators which is the sequence of maximum likelihood estimators computed on a subsample.

Value

onestep returns an object of class "onestep" inheriting from "fitdist" So, it is a list with the following components:

estimate	the parameter estimates.
method	the character string coding for the fitting method : "closed formula" for closed-form MLE or closed-form one-step, "numeric" for numeric computation of the one-step estimation.
sd	the estimated standard errors, NA if numerically not computable or NULL if not available.
cor	the estimated correlation matrix, NA if numerically not computable or NULL if not available.
vcov	the estimated variance-covariance matrix, NULL if not available.
loglik	the log-likelihood.
aic	the Akaike information criterion.
bic	the the so-called BIC or SBC (Schwarz Bayesian criterion).
n	the length of the data set.
data	the data set.
distname	the name of the distribution.
dots	the list of further arguments passed in ... to be used .
convergence	an integer code for the convergence: 0 indicates successful convergence (from explicit formula or not). 10 indicates an error.
discrete	the input argument or the automatic definition by the function to be passed to functions gofstat, plotdist and cdfcomp.
weights	the vector of weigths used in the estimation process or NULL.

Generic functions inheriting from "fitdist" objects:

print The print of a "onestep" object shows few traces about the fitting method and the fitted distribution.

summary The summary provides the parameter estimates of the fitted distribution, the log-likelihood, AIC and BIC statistics and when the maximum likelihood is used, the standard errors of the parameter estimates and the correlation matrix between parameter estimates.

plot The plot of an object of class "onestep" returned by fitdist uses the function plotdist. An object of class "onestep" or a list of objects of class "onestep" corresponding to various fits using the same data set may also be plotted using a cdf plot (function cdfcomp), a density plot(function denscomp), a density Q-Q plot (function qqcomp), or a P-P plot (function ppcomp).

logLik Extracts the estimated log-likelihood from the "onestep" object.

vcov Extracts the estimated var-covariance matrix from the "onestep" object.

coef Extracts the fitted coefficients from the "onestep" object.

Author(s)

Alexandre Brouste, Darel Noutsa Mieniedou, Christophe Dutang

References

L. Le Cam (1956). *On the asymptotic theory of estimation and testing hypothesis*. In: Proceedings of 3rd Berkeley Symposium I, pages 355–368.

K. Kamatani and M. Uchida (2015). *Hybrid multi-step estimators for stochastic differential equations based on sampled data*. Stat Inference Stoch Process, 18(2), pages 177–204.

See Also

See Also as `mledist` and `fitdist` in fitdistrplus.

Examples

```
n <- 1000

#Gamma

theta <- c(2, 3)
o.sample <- rgamma(n, shape=theta[1], rate=theta[2])
onestep(o.sample, "gamma")

#Beta

theta <- c(0.5, 1.5)
o.sample <- rbeta(n, shape1=theta[1], shape2=theta[2])
onestep(o.sample, "beta")

#Negative Binomial

theta <- c(1, 5)
o.sample <- rnbinom(n, size=theta[1], mu=theta[2])
onestep(o.sample, "nbinom")


#Cauchy

theta <- c(2, 3)
o.sample <- rcauchy(n, location=theta[1], scale=theta[2])
onestep(o.sample, "cauchy", control = list(param_t=0.3))

#Generic (dweibull2)

theta <- c(0.8, 3)
dweibull2 <- function(x, shape, scale, log=FALSE)
  dweibull(x = x, shape = shape, scale = scale, log = log)

o.sample <- rweibull(n, shape = theta[1], scale = 1/theta[2])
onestep(o.sample,  "weibull2",  method="numeric",  start=list(shape=1,  scale=1))
```

Index