Package ‘OneStep’

October 17, 2024

Type Package

Title One-Step Estimation

Version 0.9.4

Contact Alexandre Brouste, Christophe Dutang <onestep@univ-lemans.fr>

Description Provide principally an eponymic function that numerically computes the Le Cam's one-step estimator for an independent and identically distributed sample. One-step estimation is asymptotically efficient (see L. Le Cam (1956) <https://projecteuclid.org/euclid.bsmsp/1200501652>) and can be computed faster than the maximum likelihood estimator for large observation samples, see e.g. Brouste et al. (2021) <doi:10.32614/RJ-2021-044>.

License GPL (>= 2)

Encoding UTF-8

Depends fitdistrplus, numDeriv, parallel, extraDistr

Suggests actuar

URL https://journal.r-project.org/archive/2021/RJ-2021-044/

Classification/MSC-2010 62F10

NeedsCompilation no

Author Alexandre Brouste [aut] (<https://orcid.org/0000-0001-6719-7432>), Christophe Dutang [aut, cre] (<https://orcid.org/0000-0001-6732-1501>), Darel Noutsa Mieniedou [ctb]

Maintainer Christophe Dutang <dutangc@gmail.com>

Repository CRAN

Date/Publication 2024-10-17 16:50:09 UTC

Contents

`OneStep-package` *One-Step Estimation*

Description

Provide principally an eponymic function that numerically computes the Le Cam's one-step estimator for an independent and identically distributed sample. One-step estimation is asymptotically efficient (see L. Le Cam (1956) <https://projecteuclid.org/euclid.bsmsp/1200501652>) and can be computed faster than the maximum likelihood estimator for large observation samples, see e.g. Brouste et al. (2021) <doi:10.32614/RJ-2021-044>.

Details

The DESCRIPTION file:

Package:	OneStep
Type:	Package
Title:	One-Step Estimation
Version:	0.9.4
Authors@R:	c(person("Alexandre", "Brouste", role = "aut", email = "alexandre.brouste@univ-lemans.fr", com
Contact:	Alexandre Brouste, Christophe Dutang <onestep@univ-lemans.fr>
Description:	Provide principally an eponymic function that numerically computes the Le Cam's one-step estim
License:	GPL (>= 2)
Encoding:	UTF-8
Depends:	fitdistrplus, numDeriv, parallel, extraDistr
Suggests:	actuar
URL:	https://journal.r-project.org/archive/2021/RJ-2021-044/
Classification/MSC-2010:	62F10
Author:	Alexandre Brouste [aut] (<https://orcid.org/0000-0001-6719-7432>), Christophe Dutang [aut, cre
Maintainer:	Christophe Dutang <dutangc@gmail.com>

Index of help topics:

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

Author(s)

Alexandre Brouste [aut] (<https://orcid.org/0000-0001-6719-7432>), Christophe Dutang [aut, cre] (<https://orcid.org/0000-0001-6732-1501>), Darel Noutsa Mieniedou [ctb]

Maintainer: Christophe Dutang <dutangc@gmail.com>

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

`benchonestep` performs a benchmark of one-step MLE against other methods on a given dataset. `benchonestep.replicate` repeats several times the procedure: data random generation and benchmark through `benchonestep`.

Usage

```
benchonestep(data, distr, methods, init, weights=NULL,...)
benchonestep.replicate(nsample, nbsimu, distr, methods=NULL, echo=FALSE, ncpus=1, ...)
```

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 vector of methods: character among `"mme"`, `"mle"`, `"onestep"` (can be abbreviated).
`init`	A named list for the intial guess method.
`weights`	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 for `benchonestep`; true parameters passed to `rdistr` for `benchonestep.replicate`
`nsample`	a numeric for the sample size.
`nbsimu`	a numeric for the replication number.
`echo`	a logical to display or not some traces of benchmarking.
`ncpus`	Number of processes to be used in parallel operation: typically one would fix it to the number of available CPUs.

Value

A matrix with estimate and time in seconds per method for `benchonestep`; an array with estimates, times, errors in seconds per method, per simulation for `benchonestep.replicate`.

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.

Examples

```
n <- 1000
set.seed(1234)

x <- rbeta(n, 3, 2)
benchonestep(x, "beta", c("mle", "one"))
```

`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 named list for the user initial guess estimation.
`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`.

`nbstep` the number of Newton step, 0 for closed-form MLE, 1 for one-step estimators and 2 for two-step estimators.

`delta` delta parameter (used for sub-sample guess estimator).

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, Christophe Dutang, Darel Noutsa Mieniedou

References

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

I.A. Koutrouvelis (1982). *Estimation of Location and Scale in Cauchy Distributions Using the Empirical Characteristic Function*, Biometrika, 69(1), 205-213.

U. Grenander (1965). *Some direct estimates of the mode*, Annals of Mathematical Statistics, 36, 131-138.

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

Z.-S. Ye and N. Chen (2017). *Closed-Form Estimators for the Gamma Distribution Derived From Likelihood Equations*, The American Statistician, 71(2), 177-181.

See Also

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

Examples

```
n <- 1000
set.seed(1234)

##1. Gamma

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

#Default method
onestep(o.sample, "gamma")

#User initial sequence of guess estimator
# See : Ye and Chen (2017)

qtmp <- sum(o.sample*log(o.sample))-sum(log(o.sample))*mean(o.sample)
alphastar <- sum(o.sample)/qtmp
betastar <- qtmp/length(o.sample)
thetastar <- list(shape=alphastar,rate=1/betastar)

onestep(o.sample, "gamma", init=thetastar)

#Numerical method (for comparison)
onestep(o.sample, "gamma", method="numeric")

##2.Beta

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

##3. Cauchy

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

#User initial sequence of guess estimator
#See Koutrouvelis (1982).

t <- 1/4
Phi <- mean(exp(1i*t*o.sample))
S <- Re(Phi)
Z <- Im(Phi)
thetastar <- list(location=atan(Z/S)/t,scale=-log(sqrt(S^2+Z^2))/t)
onestep(o.sample, "cauchy", init=thetastar)


##Chi2

theta <- 5
o.sample <- rchisq(n,df=theta)
```

```
onestep(o.sample,"chisq")

#User initial sequence of guess estimator
#See Grenander (1965).

p <- n^(2/7)
k <- floor(n^(3/5))
Dstar <- sort(o.sample)
Dk1 <- Dstar[(1+k):n]
Dk2 <- Dstar[1:(n-k)]
sigma.star <- 1/2*sum((Dk1+Dk2)*(Dk1-Dk2)^(-p))/sum((Dk1-Dk2)^(-p))+2

onestep(o.sample,"chisq",init=list(df=sigma.star))


#Negative Binomial

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


#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",
  init=list(shape=1,  scale=1))
```

Index