Package 'bootBCa'

July 3, 2014

Type Package

Title Function to find nonparametric BCa intervals

Version 1.0

Date 2014-07-03

Author S original, from StatLib, by Rob Tibshirani. R port by
Friedrich Leisch. Enhancements by David Flater.

Maintainer David Flater <dflater@nist.gov>

Description Function BCa finds confidence intervals using Efron's
nonparametric bias-corrected and accelerated (BCa) bootstrap method. It is
an enhanced derivative of the function bcanon, forked from
bootstrap_2012.04-1. Adaptive determination of the number of bootstrap
replications is supported and the amount of memory required is less by a factor of nboot.

License BSD_3_clause + file LICENSE

NeedsCompilation no

Depends stats, R (>= 2.10.0)

R topics documented:

BCa 1

Index 7

`BCa` *Function to find nonparametric BCa intervals*

Description

The package bootBCa exports only a single function, BCa.

Function BCa finds confidence intervals using Efron's nonparametric bias-corrected and accelerated (BC_a) bootstrap method.

Function BCa was based on function bcanon in the R package bootstrap_2012.04-1 but was significantly enhanced. Functional differences include:

1. Implement adaptive bootstrap.
2. Reduce memory required by a factor of nboot.
3. Utilize the quantile function from stats, replacing code that contained an off-by-one error.*
4. Add option to use different types of quantiles.
5. Eliminate superfluous returns; add returns for bootstrap estimate, attained precision, and the number of bootstrap replications done.
6. Change default alpha to just c(0.025,0.975).
7. Add some error checks.
8. If all values in x are the same, skip the bootstrap and return theta(x,...) for all requested quantiles.**

* This bug persists in bootstrap_2014.4 but has been reported to the new maintainer and should be fixed in the next version.

** In this degenerate case, boot_1.3-9 crashes and bootstrap_2012.04-1 returns NAs.

Usage

```
BCa(x, delta, theta, ..., alpha = c(0.025, 0.975), verbose = F,
    M = 25000, Mlimit = 1500000, qtype = 1)
```

Arguments

`x`	The data (vector).
`delta`	Requested precision. The number of bootstrap replications is adapted dynamically to deliver the requested precision using the procedure described in JCGM 101:2008 section 7.9.4. Beware, the number of replications needed grows explosively if delta is too small. If delta is NA, just do M replications (non-adaptive).
`theta`	Function returning the statistic of interest (e.g., mean).
`...`	Additional arguments for theta (optional).
`alpha`	Vector of quantiles for the desired BC_a confidence intervals.
`verbose`	Set verbose to T to get a progress report on the adaptive bootstrap process and other details as the BCa function runs.
`M`	Bootstrap replication batch size. Adaptive does at least 2M replications before it can terminate.
`Mlimit`	Stop increasing the number of bootstrap replications if it takes more than this number to make delta. When delta is NA this parameter is irrelevant.
`qtype`	Use this type of quantile. The value of qtype is an integer between 1 and 9 selecting one of the nine quantile algorithms implemented by the quantile function in stats. The default, type 1, is the inverse of the empirical distribution function with no interpolation between values and is closest to what was originally implemented in bcanon. See also Hyndman and Fan.

Value

The output is a vector:

`[1], "nboot"` Number of bootstrap replications done.

`[2], "prec"` Estimate of attained precision. This can exceed delta if the iterations limit Mlimit is reached before the requested precision is achieved. For non-adaptive, this is NA or 0.

`[3], "est"` Bootstrap estimate (mean of the $\hat{\theta}^*$ values).

`rest, "0.xxx"`
Points corresponding to alpha values.

Stop conditions and special cases

1. Most invalid parameter values result in an immediate stop.
2. If theta returns a non-scalar value (i.e., a value of length not equal to 1), the function will stop with "BCa: theta returned a non-scalar value."
3. If all values in x are the same, including if x is of length 1, no bootstrapping is done. Instead, zero is returned for the number of bootstrap replications and the estimate of attained precision, and theta(x,...) is returned for the bootstrap estimate and all requested quantiles. In this special case, theta(x,...) is allowed to be non-numeric, infinite, or NaN.
4. In all other cases, if theta returns a non-finite or non-numeric value, including NA or NaN, the function will stop with "theta returned a non-(finite,numeric) value."
5. If the value of theta is always the same after removing any single element from x, the function will stop with "acceleration is indeterminate for insensitive function." In such cases, other kinds of bootstrap intervals (not implemented in this package) can still be used.
6. The function will stop with "BCa: alpha value too extreme for these data" if an alpha value is so close to 0 or 1 that the BC_a interval becomes invalid.

Limitations of the BCa function

There is no support for doing multiple statistics simultaneously, nor does the function parallelize internally. However, multiple invocations of the function can profitably be run in parallel on different data (e.g., different treatments of an experiment).

As described in JCGM 101:2008 section 7.9.4, the convergence criterion for the adaptive method corresponds to a coverage probability of approximately 95 %. In other words, even when the returned estimate of attained precision is numerically less than delta and the number of bootstrap replications done was not limited by Mlimit, there is approximately a 5 % probability that the precision attained was actually worse than delta (numerically greater than delta).

Stopping when theta returns infinity is overkill in the non-adaptive case where an infinity in the set of $\hat{\theta}^*$ values might merely cause the bootstrap estimate to become infinite. On the other hand, an infinity arising during the jacknife is fatal, even in non-adaptive, so continuing with business as usual after the theta function has been demonstrated to return infinities is arguably masking a fault.

Contraindications for use of the BC_a method

Like the ordinary percentile interval, the BC_a interval becomes inaccurate when the number of bootstrap replications is not much larger than $1/\min(\alpha, 1-\alpha)$ because there are not enough $\hat{\theta}^*$ values from which to extract the desired quantile (e.g., one cannot plausibly estimate a 99 % quantile given fewer than 100 values). However, the bias correction and acceleration adjustments made in BC_a

can result in a more extreme quantile being needed. Therefore, BC_a intervals are contraindicated if it is not possible to use a generously large value of M.

Although it depends on the distribution, the empirical error rate of BC_a intervals tends to significantly exceed the nominal error rate when the sample size is very small (on the order of 10). Examples are given in Davison and Hinkley and Carpenter and Bithell.

Carpenter and Bithell show that the coverage error for BC_a increases as alpha values approach 0 or 1. Specifically, Efron's function

$$z_0 + \frac{z_0 + z^{(\alpha)}}{1 - a\left(z_0 + z^{(\alpha)}\right)}$$

approaches $z_0 - 1/a$ instead of infinity as $z^{(\alpha)}$ becomes infinite. There is thus concern about coverage error when alpha values become "extreme" in context of the acceleration and bias estimates arising from the data provided. In the maintainer's experience, with non-contrived data, this particular error remains insignificant for all practically useful values of alpha. Unequivocally, however, Efron's function has a pole where $z_0 + z^{(\alpha)} = 1/a$ and produces invalid results when $a\left(z_0 + z^{(\alpha)}\right) \geq 1$. The BCa function stops with "BCa: alpha value too extreme for these data" when this condition arises.

If the theta function returns few distinct values, whether due to a small or homogeneous sample or an intrinsic property of theta, the bias correction and consequently the BC_a interval become erratic. For example, if x=c(1,2) and theta=mean, only three distinct values are returned (25 % 1, 50 % 1.5, 25 % 2), z_0 is forced away from zero, and the intervals produced by BC_a are skewed. In this case the simpler methods that do not attempt to correct for bias are more reliable:

```
library('boot')
boot.ci(boot.out=boot(data=c(1,2),statistic=function(x,index)
mean(x[index]),R=10000),type=c("norm","basic","perc","bca"),
conf=seq(0.7,0.9,0.1))

Intervals :
Level      Normal              Basic
70%   ( 1.135,  1.863 )   ( 1.000,  2.000 )
80%   ( 1.050,  1.948 )   ( 1.000,  2.000 )
90%   ( 0.922,  2.076 )   ( 1.000,  2.000 )

Level     Percentile            BCa
70%   ( 1.0,  2.0 )   ( 1.0,  1.5 )
80%   ( 1.0,  2.0 )   ( 1.0,  1.5 )
90%   ( 1.0,  2.0 )   ( 1.0,  1.5 )
```

Acknowledgment

Thanks to William F. Guthrie of NIST's Statistical Engineering Division for helpful reviews and support.

License

This package was based on function bcanon in the "ORPHANED" R package bootstrap_2012.04-1 from http://cran.r-project.org/web/packages/bootstrap/ but was significantly enhanced. These enhancements were made by David Flater <dflater@nist.gov> in the course of official duties. Pursuant to Title 17 Section 105 of the United States Code, the enhancements are not subject to copyright protection and are in the public domain. To the extent that this is a derivative work, however, the previous copyrights would apply.

The R bootstrap package rev. 2012.04-1 was ported from StatLib in 2000 by Friedrich Leisch and last modified by Kjetil Halvorsen in 2012. Its LICENSE file says:

```
YEAR: 2000
COPYRIGHT HOLDER: Rob Tibshirani, Friedrich Leisch
ORGANIZATION: Stanford University
```

Its DESCRIPTION file says: `License: BSD_3_clause + file LICENSE`.

The referenced BSD_3_clause is the following (from http://cran.r-project.org/web/licenses/BSD_3_clause):

```
Copyright (c) <YEAR>, <COPYRIGHT HOLDER>

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

    Redistributions of source code must retain the above copyright
    notice, this list of conditions and the following disclaimer.

    Redistributions in binary form must reproduce the above copyright
    notice, this list of conditions and the following disclaimer in
    the documentation and/or other materials provided with the
    distribution.

    Neither the name of the <ORGANIZATION> nor the names of its
    contributors may be used to endorse or promote products derived
    from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

Subsequently, maintenance of the bootstrap package was resumed by Scott Kostyshak (rev. 2014.4).

The S StatLib package was by Rob Tibshirani to accompany the book *An Introduction to the Bootstrap* by Bradley Efron and Robert J. Tibshirani, 1993.

NIST assumes no responsibility for use of this software by other parties and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic.

Author(s)

S original, from StatLib, by Rob Tibshirani. R port by Friedrich Leisch. Enhancements by David Flater.

Maintainer: David Flater <dflater@nist.gov>

References

Bradley Efron. Better bootstrap confidence intervals. Journal of the American Statistical Association, 82(397):171–185, March 1987. http://www.jstor.org/stable/2289144. See also the comments and rejoinder that follow on pages 186–200, http://www.jstor.org/stable/i314281.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

Thomas J. DiCiccio and Bradley Efron. Bootstrap Confidence Intervals. Statistical Science, 11(3):189–212, August 1996. http://www.jstor.org/stable/2246110. See also the comments and rejoinder that follow on pages 212–228, http://www.jstor.org/stable/i313095.

Joint Committee for Guides in Metrology. Evaluation of measurement data—Supplement 1 to the "Guide to the expression of uncertainty in measurement"—Propagation of distributions using a Monte Carlo method. JCGM 101:2008, http://www.bipm.org/utils/common/documents/jcgm/JCGM_101_2008_E.pdf.

Rob J. Hyndman and Yanan Fan. Sample Quantiles in Statistical Packages. The American Statistician, 50(4):361–365, November 1996. http://www.jstor.org/stable/2684934.

Anthony C. Davison and David V. Hinkley. *Bootstrap Methods and their Application*. Cambridge University Press, 1997.

James Carpenter and John Bithell. Bootstrap confidence intervals: when, which, what? A practical guide for medical statisticians. Statistics in Medicine, 19(9):1141–1164, May 2000. http://dx.doi.org/10.1002/(SICI)1097-0258(20000515)19:9<1141::AID-SIM479>3.0.CO;2-F.

See Also

bootstrap, boot.

Examples

```
data <- rnorm(n=1000,mean=20,sd=5)

# 95 % confidence interval of mean with an adaptive replication count.
print(BCa(data,0.01,mean))

# 90 % confidence interval of mean, non-adaptive replication count.
print(BCa(data,NA,mean,alpha=c(0.05,0.95),M=10000))

# User-defined function with an arbitrary extra parameter.
fudgedmean <- function(x, fudge_factor) {
  mean(x) + fudge_factor
}
print(BCa(data,0.01,fudgedmean,3))

# Degenerate case: 0 0 0 0 0
print(BCa(rep(5,100),0.01,sd))

# Degenerate case, single sample: 0 0 NA NA NA
print(BCa(5,0.01,sd))
```

Index

∗Topic **nonparametric**
BCa, 1
∗Topic **package**
BCa, 1
∗Topic **robust**
BCa, 1

BCa, 1
bcanon, *2*
boot, *6*
bootBCa-package *(BCa)*, 1
bootstrap, *6*

quantile, *2*

stats, *2*