Package ‘logconcens’
February 20, 2015

Type Package
Title Maximum likelihood estimation of a log-concave density based on
censored data
Version 0.16-4
Date 2013-12-13
Author Dominic Schuhmacher <dominic.schuhmacher@mathematik.uni-goettingen.de>, Kaspar Rufibach <kaspar.rufibach@gmail.com>, Lutz Duembgen <duembgen@stat.unibe.ch>
Maintainer Dominic Schuhmacher
<dominic.schuhmacher@mathematik.uni-goettingen.de>
Depends R (>= 2.15.0), base, methods, utils, grDevices, graphics,
stats
Suggests logcondens
Description Based on right or interval censored data, compute the maximum likelihood estima-
tor of a (sub)probability density under the assumption that it is log-concave. For further informa-
License GPL (>= 2)
URL http://www.dominic.schuhmacher.name, http://www.kasparrufibach.ch,
http://www.staff.unibe.ch/duembgen
LazyLoad yes
NeedsCompilation yes
Repository CRAN
Date/Publication 2013-12-13 15:55:21

R topics documented:

logconcens-package .................................................. 2
cure.profile .......................................................... 3
c.l.control ............................................................. 5
c.ldensity-methods .................................................. 6
c.logcon ............................................................... 8
c.loglike ............................................................... 14
plotint ............................................................... 15
**Description**

Based on independent intervals $X_i = [L_i, R_i]$, where $-\infty < L_i \leq R_i \leq \infty$, compute the maximum likelihood estimator of a (sub)probability density under the assumption that it is log-concave. For further information see Duembgen, Rufibach, and Schuhmacher (2013, preprint).

**Details**

- **Package**: logconcens
- **Type**: Package
- **Version**: 0.16-4
- **Date**: 2013-12-13
- **License**: GPL (>=2)
- **LazyLoad**: yes

The main function is `logcon`, which offers computation of the MLE for many types of censored and also exact data. Various parameters can be set that allow for fine control of the underlying EM algorithm in “difficult” situations. An object of type `lcdensity` is returned, for which `plot`, `print`, and `summary` methods are available. There is also a function `loglike` for computing the log-likelihood of a `lcdensity` object.

**Author(s)**

Dominic Schuhmacher <dominic.schuhmacher@mathematik.uni-goettingen.de>
Kaspar Rufibach <kaspar.rufibach@gmail.com>
Lutz Duembgen <duembgen@stat.unibe.ch>

Maintainer: Dominic Schuhmacher <dominic.schuhmacher@mathematik.uni-goettingen.de>

**References**


**Examples**

```r
## Simple examples with simulated data.
## For more detailed examples see the help for the function logcon.
```
## Cure Profile

Evaluate the profile log-likelihood on a grid of $p_0$-values.

### Description

For each of a series of values for the cure parameter $p_0$ run the function `logcon` and evaluate the (normalized) log-likelihood at $(\phi, p_0)$, where $\phi$ is the log subprobability density returned by `logcon`. This serves for (approximate) joint likelihood maximization in $(\phi, p_0)$.

### Usage

```R
cure.profile(x, p0grid=seq(0,0.95,0.05), knot.prec=IQR(x[x<Inf])/75, reduce=TRUE, control=lc.control())
```
Arguments

- \( x \) a two-column matrix of \( n \geq 2 \) rows containing the data intervals.
- \( p_0 \text{grid} \) a vector of values \( p_0 \) for which the profile log-likelihood is to be evaluated.
- knot.prec, reduce, control arguments passed to the function \texttt{logcon}.

Value

A list containing the following values:

- \( p_0 \hat{\text{hat}} \) the element in \( p_0 \text{grid} \) that maximizes the profile likelihood (in the very unlikely case of ties, only the smallest such element is returned).
- \( \text{status} \) the vector of (normalized) profile log-likelihood values for the elements of \( p_0 \text{grid} \).

Note

For a large \( p_0 \text{grid} \)-vector (fine grid) computations may take a long time. Consider using the option adapt.p0 in the function \texttt{logcon} for a much faster method of joint likelihood maximization in \((\phi,p_0)\).

Author(s)

Dominic Schuhmacher <dominic.schuhmacher@mathematik.uni-goettingen.de>
Kaspar Rufibach <kaspar.rufibach@gmail.com>
Lutz Duembgen <duembgen@stat.unibe.ch>

See Also

\texttt{logcon}, \texttt{loglike}

Examples

```r
## The example from the logconcens-package help page:
set.seed(11)
x <- rgamma(50,3,1)
x <- cbind(x,ifelse(rexp(50,1/3) < x,Inf,x))

## Not run:
plotint(x)
progrid <- seq(0.1,0.6,0.025)
prores <- cure.profile(x, progrid)
plot(progrid, prores$loglike)
prores$p0hat
res <- logcon(x, p0=prores$p0hat)
plot(res, type="survival")

## End(Not run)
```
lc.control

Set the control parameters for logcon.

Description

Allows to set the control parameters for the more technical aspects of the function logcon and provides default values for any parameters that are not set.

Usage

lc.control(maxiter=49, move.prec=1e-5, domind1l=1, domind2r=1, force.inf=FALSE, red.thresh=NULL, check.red=TRUE, addpoints=FALSE, addeps=NULL, preweights=NULL, minw=0, show=FALSE, verbose=FALSE)

Arguments

maxiter the maximal number of iterations in the main EM algorithm. Default is 49 rather than 50, because this goes well with plotting in case of show = TRUE.

move.prec a real number giving the threshold for the $L_1$-distance between densities in subsequent steps below which the algorithm is stopped.

domind1l, domind2r index numbers in the vector of sorted interval endpoints that specify the left and right boundary of the maximal domain to be considered by the algorithm; see the details section of the help for logcon. The indices are counted from the left and from the right, respectively. So the default values of domind1l = 1 and domind2r = 1 mean that the largest possible domain is used.

force.inf logical. For experimental use only. Should the domain interval be forced to be right-infinite (if there is a right-infinite data interval)?

red.thresh a real number indicating the threshold below which the boundary integrals are considered too small; see the details section of the help for logcon. There is a sensible default, which depends on check.red.

check.red logical. If a boundary integral is deemed too small, should the derivative of the augmented log-likelihood be checked to confirm the domain reduction.

addpoints logical. Should extra exact observations be added to the data at the left- and rightmost finite interval endpoints to prevent domain reduction? These observations obtain a small weight < 1 as compared to the weight of 1 for all the other observation intervals. The weight is specified by addeps.

addeps a positive real number. If NULL, a default value of $1/n^2$ is computed where $n$ is the number of observation intervals. See addpoints.

preweights a vector of weights for the observation intervals. Defaults to rep(1,n).

minw a positive real number. This gives another way for preventing domain reduction. Instead of adding observations the weights for the internal active set algorithm are kept at or above minw at the boundary of the domain.
show logical. Should progress of the algorithm be plotted? Warning: if TRUE, this may open many new graphics devices in case of complicated data sets.

verbose logical. Should additional information about the progress of the algorithm be printed? This mainly prints quantities important for the decision to reduce the domain of the function and about the progress of the EM algorithm.

Details

For further explanations about the algorithm see the help for logcon. In summary:

maxiter and move.prec provide stopping criteria for the EM algorithm.

domind11, domind2r, force.inf, red.thresh, and check.red control aspects related to domain reduction.

addpoints, addeps, preweights, winw allow for reweighing of data interval, mainly for increasing numerical stability by preventing domain reduction.

show and verbose give illustrations and background information of the run of the algorithm.

Value

A list with all of the above components set to their (specified or default) value.

Author(s)

Dominic Schuhmacher <dominic.schuhmacher@mathematik.uni-goettingen.de>
Kaspar Rufibach <kaspar.rufibach@gmail.com>
Lutz Duembgen <duembgen@stat.unibe.ch>

See Also

logcon

Examples

## See the examples for logcon

lcdensity-methods  Methods for objects of classlcdensity.

Description

Plot, print, and summary methods for objects of classlcdensity.
Usage

## S3 method for class 'lcdensity'
plot(x, type = c("log-density", "density", "CDF", "survival"), sloperange = TRUE, kinklines=TRUE, kinkpoints=FALSE, xlim=NULL, ylim=NULL, ...)

## S3 method for class 'lcdensity'
print(x, ...)

## S3 method for class 'lcdensity'
summary(object, ...)

Arguments

- **x**, **object**: objects of class `lcdensity`, as returned by `logcon`.
- **type**: the type of plot to be produced.
- **sloperange**: logical. In cases where the cure parameter / the right-hand side slope of the log-subdensity $\phi$ is not unique, should grey area be drawn indicating the set of possible right-hand slopes?
- **kinklines**: logical. Should vertical lines be drawn at the kinks of the log-subdensity $\phi$?
- **kinkpoints**: logical. Should fat points be plotted at the kinks of the log-subdensity $\phi$?
- **xlim**, **ylim**: numeric vectors of length 2, giving the x and y coordinates ranges.
- **...**: further arguments passed to `plot.default`. Depending on the argument this may or may not work in the intended way.

Author(s)

Dominic Schuhmacher <dominic.schuhmacher@mathematik.uni-goettingen.de>
Kaspar Rufibach <kaspar.rufibach@gmail.com>
Lutz Duembgen <duembgen@stat.unibe.ch>

See Also

- `plotint`

Examples

## See the examples for logcon
Compute log-concave MLE based on censored or exact data.

Description

Based on independent intervals $X_i = [L_i, R_i]$, where $-\infty < L_i \leq R_i \leq \infty$, compute the maximum likelihood estimator of a (sub)probability density $\phi$ and the remaining mass $p_0$ at infinity (also known as cure parameter) under the assumption that the former is log-concave. Computation is based on an EM algorithm. For further information see Dümbgen, Rufibach, and Schuhmacher (2013, preprint).

Usage

- `logcon(x, adapt.p0=FALSE, p0=0, knot.prec=IQR(x[x<Inf])/75, reduce=TRUE, control=lc.control())`
- `logConCens(x, adapt.p0=FALSE, p0=0, knot.prec=IQR(x[x<Inf])/75, reduce=TRUE, control=lc.control())`
- `logconcure(x, p0=0, knot.prec=IQR(x[x<Inf])/75, reduce=TRUE, control=lc.control())`

Arguments

- **x**: a two-column matrix of $n \geq 2$ rows containing the data intervals, or a vector of length $n \geq 2$ containing the exact data points.
- **adapt.p0**: logical. Should the algorithm be allowed to adapt $p_0$? In this case an alternating maximization procedure is used that is believed to always yield a joint maximizer $(\hat{\phi}, \hat{p}_0)$. For the much slower (but maybe safer) profile likelihood maximization method, see the function `cure.profile`.
- **p0**: a number from 0 to 1 specifying the mass at infinity. If the algorithm is allowed to adapt $p_0$, this argument only specifies the starting value. Otherwise it is assumed that the true cure parameter $p_0$ is equal to this number. In particular, for the default setting of 0, a proper probability density $\phi$ is estimated.
- **knot.prec**: the maximal distance between two consecutive grid points, where knots (points at which the resulting log-subdensity $\phi$ may change slope) can be positioned. See details.
- **reduce**: logical. Should the domain of the (sub)density be reduced whenever the mass at the left or the right boundary becomes too small?
- **control**: a list of control parameters for the more technical aspects of the algorithm; usually the result of a call to `lc.control`.

Details

Based on the data intervals $X_i = [L_i, R_i]$ described above, function `logcon` computes a concave, piecewise linear function $\phi$ and a probability $p_0$ which satisfy $\int \exp \phi(x) \, dx = 1 - p_0$ and jointly maximize the (normalized) log-likelihood.
\[ \ell(\phi, p_0) = \frac{1}{n} \sum_{i=1}^{n} \left[ 1\{L_i = R_i\} \phi(X_i) + 1\{L_i < R_i\} \log \left( \int_{L_i}^{R_i} \exp \phi(x) \, dx + 1\{R_i = \infty\}p_0 \right) \right]. \]

If \( x \) is a two-column matrix, it is assumed to contain the left and right interval endpoints in the correct order. Intervals may have length zero (both endpoints equal) or be unbounded to the right (right endpoint is \( \infty \)). Computation is based on an EM algorithm, where the M-step uses an active set algorithm for computing the log-concave MLE for exact data with weights. The active set algorithm was described in Duembgen, Huesler, and Rufibach (2007) and Duembgen and Rufibach (2011) and is available in the R package \texttt{logcondens}. It has been re-implemented in C for the current package because of speed requirements. The whole algorithm for censored data has been indicated in Duembgen, Huesler, and Rufibach (2007) and was elaborated in Duembgen, Schuhmacher, and Rufibach (2013, preprint).

If \( x \) is a vector argument, it is assumed to contain the exact data points. In this case the active set algorithm is accessed directly.

In order to obtain a finite dimensional optimization problem the (supposed) domain of \( \phi \) is subdivided by a grid. Stretches between interval endpoints where for theoretical reasons no knots (points where the slope of \( \phi \) changes) can lie are left out. The argument \( \text{kink.prec} \) gives the maximal distance we allow between consecutive grid points in stretches where knots can lie. Say \( \text{plotint}(x) \) to see the grid.

The EM algorithm works only for fixed dimensionality of the problem, but the domain of the function \( \phi \) is not a priori known. Therefore there is an outer loop starting with the largest possible domain, given by the minimal and maximal endpoints of all the intervals, and decreasing the domain as soon as the EM steps let \( \phi \) become very small towards the boundary. “Very small” means that the integral of \( \exp \circ \phi \) over the first or last stretch between interval endpoints within the current domain falls below a certain threshold \( \text{red.thresh} \), which can be controlled via \texttt{lc.control}.

Domain reduction tends to be rather conservative. If the computed solution has a suspiciously steep slope at any of the domain boundaries, the recommended strategy is to enforce a smaller domain by increasing the parameters \( \text{domind1} \) and/or \( \text{domind2r} \) via \texttt{lc.control}. The function \texttt{loglike} may be used to compare the (normalized) log-likelihoods of the results.

\texttt{logConCens} is an alias for \texttt{logcon}. It is introduced to provide unified naming with the main functions in the packages \texttt{logcondens} and \texttt{logcondiscr}.

\texttt{logconcure} is the same as \texttt{logcon} with \( \text{adapt.p0} = \text{TRUE} \) fixed.

### Value

An object of class \texttt{lcdensity} for which reasonable \texttt{plot}, \texttt{print}, and \texttt{summary} methods are available.

If the argument \( x \) is a two-column matrix (censored data case), such an object has the following components.

- \texttt{basedon} the string "censored" for the type of data the solution is based on.
- \texttt{status} currently only \( 0 \) if the algorithm converged; and \( 1 \) otherwise. Note that in most cases even with status \( 1 \) the returned solution is very close to the truth. The \( 1 \) is often due to the fact that the termination criterion is not so well balanced yet.
x the data entered.

tau the ordered vector of different interval endpoints.

domind1, domind2 the indices of the tau-element at which the domain of the MLE \( \phi \) starts/ends.

tplus the grid vector. \( \tau[\text{domind1}:\text{domind2}] \) augmented by points of subdivision.

isknot 0-1 value. For the finite elements of tplus a 1 if \( \phi \) has a knot at this position, 0 otherwise.

phi the vector of \( \phi \)-values at the finite elements of tplus.

phislr if \( \sup(\text{dom}(\phi)) = \infty \), the slope of \( \phi \) after the last knot. Otherwise \(-\infty\).

phislr.range a vector of length 2 specifying a range of possible values for phislr. This is for the (rather rare) situations that mass may be shifted between the interval from the rightmost tau-point to infinity and the cure parameter without changing the likelihood. Otherwise phislr.range is NA.

cure the cure parameter. Either the original argument \( p_0 \) if adapt.p0 was FALSE, otheriwse the estimated cure parameter obtained by the alternating maximization procedure.

cure.range a vector of length 2 specifying a range of possible values for cure or NA. See phislr.range.

Fhat the vector of values of the distribution function \( F \) of \( \exp \circ \phi \) at the finite elements of tplus.

Fhatfin the computed value of \( \lim_{t \to \infty} F(t) \).

Note

If \( x \) is a vector, this function does the same as the function \text{logConDens} in the package \text{logcondens}. The latter package offers additional features such as grid-based computation with weights (for high numerical stability) and smoothing of the estimator, as well as nicer plotting. For exact data we recommend using \text{logConDens} for everyday data analysis. \text{logcon} with a vector argument is to be preferred if time is of the essence (for data sets with several thousands of points or repeated optimization in iterative algorithms) or if an additional slope functionality is required.

Two other helpful packages for log-concave density estimation based on exact data are \text{logcondiscr} for estimating a discrete distribution and \text{LogConcDEAD} for estimating a multivariate continuous distribution.

Author(s)

Dominic Schuhmacher <dominic.schuhmacher@mathematik.uni-goettingen.de>
Kaspar Rufibach <kaspar.rufibach@gmail.com>
Lutz Duembgen <duembgen@stat.unibe.ch>

References


See Also

lc.control,lcdensity-methods,loglike

Examples

# A function for artificially censoring exact data
censor <- function(y, timemat) {
  tm <- cbind(0, timemat, Inf)
  n <- length(y)
  res <- sapply(1:n, function(i){
    return( c( max(tm[i,][tm[i,] < y[i]]), min(tm[i,][tm[i,] >= y[i]]) ) )
  })
  return(t(res))
}

# interval censored data

set.seed(20)
n <- 100
# generate exact data:
y <- rgamma(n,3)
# generate matrix of inspection times:
itimes <- matrix(rexp(10*n), n,10)
itimes <- t(apply(itimes,1,cumsum))
# transform exact data to interval data
x <- censor(y, itimes)
# plot both
plotint(x, imarks=y)

# Compute censored log-concave MLE
# (assuming only the censored data is available to us)
res <- logcon(x)
plot(res)

# Compare it to the log-concave MLE for the exact data
# and to the true Gamma(3,1) log-density
res.ex <- logcon(y)
lines(res.ex$x, res.ex$phi, lwd=2.5, lty=2)
xi <- seq(0.14, 0.05)
lines(xi,log(dgamma(xi,3,1)), col=3, lwd=2)
# censored data with cure

```r
## Not run:
set.seed(21)
n <- 100
# generate exact data:
y <- rgamma(n,3)
cured <- as.logical(rbinom(n,1,0.3))
y[cured] <- Inf

# generate matrix of inspection times:
itimes <- matrix(rexp(6*n),n,6)
itimes <- t(apply(itimes,1,cumsum))
# transform exact data to interval data
x <- censor(y, itimes)
# plot both
plotint(x, imarks=y)

# Compute censored log-concave MLE including cure parameter
# (assuming only the censored data is available to us)
res <- logcon(x, adapt.p0=TRUE)
plot(res)
# There is a trade-off between right-hand slope and cure parameter here
# (seen by the grey area on the right), but the margin is very small:
res$cure.range

# Compare the corresponding CDF to the true CDF
plot(res, type="CDF")
xi <- seq(0,14,0.05)
lines(xi,0.7*pgamma(xi,3,1), col=3, lwd=2)
# Note that the trade-off for the right-hand slope is not visible anymore
# (in terms of the CDF the effect is too small)

## End(Not run)
```

# real right censored data with cure

```r
# Look at data set ovarian from package survival
# Gives survival times in days for 26 patients with advanced ovarian carcinoma,
# ignoring the covariates

# Bring data to right format and plot it
## Not run:
library(survival)
data(ovarian)
sobj <- Surv(ovarian$futime, ovarian$fustat)
```
x <- cbind(sobj[,1], ifelse(as.logical(sobj[,2]), sobj[,1], Inf))
plotint(x)

# Compute censored log-concave MLE including cure parameter
res <- logcon(x, adapt.p0=TRUE)

# Compare the corresponding survival function to the Kaplan-Meier estimator
plot(res, type="survival")
res.km <- survfit(sobj ~ 1)
lines(res.km, lwd=1.5)
## End(Not run)

# --------------------------
# current status data
# --------------------------

## Not run:
set.seed(22)
n <- 200
# generate exact data
y <- rweibull(n, 2, 1)
# generate vector of inspection times
itime <- matrix(rexp(n), n, 1)
# transform exact data to interval data
x <- censor(y, itime)
# plot both
plotint(x, imarks=y)

# Compute censored log-concave MLE
# (assuming only the censored data is available to us)
res <- logcon(x)
plot(res, type="CDF")

# Compare it to the true Weibull(2,1) c.d.f.
xi <- seq(0.3, 0.05)
lines(xi, pweibull(xi, 2, 1), col=3, lwd=2)
## End(Not run)

# --------------------------
# rounded/binned data
# --------------------------

## Not run:
set.seed(23)
n <- 100
# generate data in [0,1] rounded to one digit
y <- round(rbeta(n, 2, 3), 1)
# bring data to right format and plot it
x <- cbind(y-0.05, y+0.05)
loglike

Compute log-likelihood for an object of class `lcdensity`.

Description

Compute the (normalized) log-likelihood for an object of class `lcdensity` as described in the details section for the function `logcon`. The main use of this function is for comparing different results from `logcon` based on different (starting) domains.

Usage

`loglike(lcd)`

Arguments

`lcd` an object of class `lcdensity`

Value

A single numeric value, the (normalized) log-likelihood.

Author(s)

Dominic Schuhmacher <dominic.schuhmacher@mathematik.uni-goettingen.de>
Kaspar Rufibach <kaspar.rufibach@gmail.com>
Lutz Duembgen <duembgen@stat.unibe.ch>

See Also

`logcon`
plotint

Examples

```r
x <- matrix(c(0, 0.5, 0.5, 1, 1.2, 3, 3), 4, 2)
res <- logcon(x)
loglike(res)
```

---

**plotint**

*Plot censored data.*

**Description**

Plot a graphical representation of censored data specified by a two-column matrix of left and right interval endpoints. The grid of potential knots used by `logcon` is also shown.

**Usage**

```r
plotint(x, knot.prec = IQR(x[x<Inf]) / 75, imarks = NULL)
```

**Arguments**

- `x` a two-column matrix of left and right endpoints of data intervals.
- `knot.prec` the maximal distance between two consecutive grid points in the depiction of the grid used by `logcon`.
- `imarks` an optional vector of “spots” to be marked by ‘x’ for the intervals.

**Value**

Used for the side effect.

**Author(s)**

Dominic Schuhmacher <dominic.schuhmacher@mathematik.uni-goettingen.de>
Kaspar Rufibach <kaspar.rufibach@gmail.com>
Lutz Dümbgen <duembgen@stat.unibe.ch>

**See Also**

- `plot.lcdensity`

**Examples**

```r
## See the examples for logcon
```
Index

*Topic package
  logconcens-package, 2

control (lc.control), 5

cure.profile, 3, 8

lc.control, 5, 8, 9, 11
lcdensity-methods, 6
logcon, 2, 4–7, 8, 14
logConCens (logcon), 8
logconcens (logconcens-package), 2
logconcens-package, 2
logconcure (logcon), 8
loglike, 2, 4, 9, 11, 14

plot, 2, 9
plot.default, 7
plot.lcdensity, 15
plot.lcdensity (lcdensity-methods), 6
plotint, 7, 15
print, 2, 9
print.lcdensity (lcdensity-methods), 6

summary, 2, 9
summary.lcdensity (lcdensity-methods), 6