

Package ‘LogConcDEAD’

January 12, 2010

Type Package

Title Log-concave Density Estimation in Arbitrary Dimensions

Version 1.4-1

Date 2009-05-11

Depends R (>= 2.3), MASS, geometry

Suggests rgl, logcondens, mvtnorm

Author Madeleine Cule, Robert Gramacy, Richard Samworth

Maintainer Madeleine Cule <mlc40@cam.ac.uk>

Description Computes a log-concave (maximum likelihood) estimator for i.i.d. data in any number of dimensions.

License LGPL

Repository CRAN

Date/Publication 2010-01-12 11:42:27

R topics documented:

LogConcDEAD-package	2
dlcd	4
dmarglcd	5
getinfolcd	6
getweights	7
interplcd	9
interpmarglcd	10
mlelcd	11
plot.LogConcDEAD	14
print.LogConcDEAD	16
rlcd	17

Index	19
--------------	-----------

LogConcDEAD-package

Computes a log-concave (maximum likelihood) estimator for i.i.d. data in any number of dimensions

Description

This package contains a function to compute the maximum likelihood estimator of a log-concave density in any number of dimensions using Shor's r -algorithm.

Functions to plot (for 1- and 2-d data), evaluate and draw samples from the maximum likelihood estimator are provided.

Details

This package contains a selection of functions for maximum likelihood estimation under the constraint of log-concavity.

`mlelcd` computes the maximum likelihood estimator (specified via its value at data points). Output is a list of class "LogConcDEAD" which is used as input to various auxiliary functions.

`dlcd` evaluates the estimated density at a particular point.

`rlcd` draws samples from the estimated density.

`interplcd` interpolates the estimated density on a grid for plotting purposes.

`dmarglcd` evaluates the estimated marginal density by integrating the estimated density over an appropriate subspace.

`interp marglcd` evaluates a marginal density estimate at equally spaced points along the axis for plotting purposes. This is done by integrating the estimated density over an appropriate subspace.

`plot.LogConcDEAD` produces plots of the maximum likelihood estimator, optionally using the `rgl` package.

`print` and `summary` methods are also available.

Note

The authors gratefully acknowledge the assistance of Lutz Duembgen at the University of Bern for his insight into the objective function in `mlelcd`.

For one dimensional data, the active set algorithm in `logcondens` is much faster.

Author(s)

Madeleine Cule (maintainer) <m1c40@cam.ac.uk>

Robert Gramacy <bobby@statslab.cam.ac.uk>

Richard Samworth <rjs57@cam.ac.uk>

References

- Barber, C.B., Dobkin, D.P., and Huhdanpaa, H.T. (1996) *The Quickhull algorithm for convex hulls* ACM Trans. on Mathematical Software, 22(4) p.469-483 <http://www.qhull.org>
- Cule, M. L. and D'Umbgen, L. (2008) On an auxiliary function for log-density estimation, University of Bern technical report. <http://arxiv.org/abs/0807.4719>
- Cule, M. L., Samworth, R. J., and Stewart, M. I. (2007) *Maximum likelihood estimation of a log-concave density*, Submitted. <http://arxiv.org/abs/0804.3989>
- Kappel, F. and Kuntsevich, A. V. (2000) *An implementation of Shor's r-algorithm* Computational Optimization and Applications 15
<http://www.uni-graz.at/imawww/kuntsevich/solvopt/>
- Shor, N. Z. (1985) *Minimization methods for nondifferentiable functions* Springer-Verlag

See Also

[logcondens](#), [rgl](#)

Examples

```
## Some simple normal data, and a few plots

x <- matrix(rnorm(200), ncol=2)
lcd <- mlelcd(x)
g <- interplcd(lcd)
par(mfrow=c(2,2), ask=TRUE)
plot(lcd, g=g, type="c")
plot(lcd, g=g, type="c", uselog=TRUE)
plot(lcd, g=g, type="i")
plot(lcd, g=g, type="i", uselog=TRUE)

## Some plots of marginal estimates
par(mfrow=c(1,1))
g.marg1 <- interpmarglcd(lcd, marg=1)
g.marg2 <- interpmarglcd(lcd, marg=2)
plot(lcd, marg=1, g.marg=g.marg1)
plot(lcd, marg=2, g.marg=g.marg2)

## generate some points from the fitted density
generated <- rlcd(100, lcd)
genmean <- mean(generated)

## evaluate the fitted density
mypoint <- c(0, 0)
dlcd(mypoint, lcd, uselog=FALSE)
mypoint <- c(10, 0)
dlcd(mypoint, lcd, uselog=FALSE)

## evaluate the marginal density
dmarglcd(0, lcd, marg=1)
dmarglcd(1, lcd, marg=2)
```

dlcd

*Evaluation of a log-concave maximum likelihood estimator at a point***Description**

This function evaluates the density function of a log-concave maximum likelihood estimator at a point or points.

Usage

```
dlcd(x, lcd, uselog=FALSE, eps=10^-10)
lcd.eval(lcd, po, uselog=FALSE, eps=10^-10)
```

Arguments

<code>x, po</code>	Point (or matrix of points) at which the maximum likelihood estimator should be evaluated
<code>lcd</code>	Object of class "LogConcDEAD" (typically output from <code>mlelcd</code>)
<code>uselog</code>	Scalar logical: should the estimator should be calculated on the log scale?
<code>eps</code>	Tolerance for numerical stability

Details

A log-concave maximum likelihood estimate \hat{f}_n is satisfies $\log \hat{f}_n = \bar{h}_y$ for some $y \in R^n$, where

$$\bar{h}_y(x) = \inf\{h(x): h \text{ concave}, h(x_i) \geq y_i \text{ for } i = 1, \dots, n\}.$$

Functions of this form may equivalently be specified by dividing C_n , the convex hull of the data into simplices C_j for $j \in J$ (triangles in 2d, tetrahedra in 3d etc), and setting

$$f(x) = \exp\{b_j^T x - \beta_j\}$$

for $x \in C_j$, and $f(x) = 0$ for $x \notin C_n$. The estimated density is zero outside the convex hull of the data.

The estimate may therefore be evaluated by finding the appropriate simplex C_j , then evaluating $\exp\{b_j^T x - \beta_j\}$ (if $x \notin C_n$, set $f(x) = 0$).

For examples, see `mlelcd`.

`lcd.eval` is deprecated, but retained for compatibility with previous versions.

Value

A vector of maximum likelihood estimate (or log maximum likelihood estimate) values, as evaluated at the points `x`.

Author(s)

Madeleine Cule <mlc40@cam.ac.uk>

Robert Gramacy

Richard Samworth

See Also

[mlelcd](#)

dmarglcd	<i>Evaluate the marginal of multivariate log-concave maximum likelihood estimators at a point</i>
----------	---

Description

Integrates the log-concave maximum likelihood estimator of multivariate data to evaluate the marginal density at a point.

Usage

```
dmarglcd(x=0, lcd, marg=1)
lcd.marg.eval (lcd, point=0, marg=1)
```

Arguments

<code>x</code> , <code>point</code>	Point (or vector of points) at which the marginal density is to be evaluated
<code>lcd</code>	Object of class "LogConcDEAD" (typically output from mlelcd)
<code>marg</code>	Which margin is required?

Details

Given a multivariate log-concave maximum likelihood estimator in the form of an object of class "LogConcDEAD", a margin `marg`, and a real-valued point `x`, this function evaluates the estimated marginal density $\hat{f}_{n,\text{marg}}(x)$, as obtained by integrating over all the other dimensions.

For examples, see [mlelcd](#).

`lcd.marg.eval` is deprecated, but retained for compatibility with previous versions.

Value

A vector containing the values of the marginal density $\hat{f}_{n,\text{marg}}$ at the points `x`.

Author(s)

Madeleine Cule <mlc40@cam.ac.uk>

Robert Gramacy

Richard Samworth

See Also

[mlelcd](#)

getinfofcd

Construct an object of class LogConcDEAD

Description

A function to construct an object of class `LogConcDEAD` from a dataset (given as a matrix) and the value of the log maximum likelihood estimator at datapoints.

Usage

```
getinfofcd(x, y, w = rep(1/length(y), length(y)), chtol = 10^-6, MinSigma = NA, Num
```

Arguments

`x` Data in R^d , in the form of an $n \times d$ numeric matrix

`y` Value of log of maximum likelihood estimator at data points

`w` Vector of weights w_i such that the computed estimator maximizes

$$\sum_{i=1}^n w_i \log f(x_i)$$

subject to the restriction that f is log-concave. The default is $\frac{1}{n}$ for all i , which corresponds to i.i.d. observations.

`chtol` Tolerance for computation of convex hull. Altering this is not recommended.

`MinSigma` Real-valued scalar giving minimum value of the objective function

`NumberOfEvaluations`

Vector containing the number of steps, number of function evaluations, and number of subgradient evaluations. If the **SolvOpt** algorithm fails, the first component will be an error code (< 0)

Details

This function is used in [mlelcd](#)

Value

An object of class "LogConcDEAD", with the following components:

x	Data copied from input (may be reordered)
w	weights copied from input (may be reordered)
logMLE	vector of the log of the maximum likelihood estimate, evaluated at the observation points
NumberOfEvaluations	Vector containing the number of steps, number of function evaluations, and number of subgradient evaluations. If the SolvOpt algorithm fails, the first component will be an error code (< 0).
MinSigma	Real-valued scalar giving minimum value of the objective function
b	matrix (see Details)
beta	vector (see Details)
triang	matrix containing final triangulation of the convex hull of the data
verts	matrix containing details of triangulation for use in dlcd
vertsoffset	matrix containing details of triangulation for use in dlcd
chull	Vector containing vertices of faces of the convex hull of the data
outnorm	matrix where each row is an outward pointing normal vectors for the faces of the convex hull of the data. The number of vectors depends on the number of faces of the convex hull.
outoffset	matrix where each row is a point on a face of the convex hull of the data. The number of vectors depends on the number of faces of the convex hull.

Author(s)

Madeleine Cule <mlc40@cam.ac.uk>

Robert B. Gramacy

Richard Samworth

See Also

[mlelcd](#)

getweights

Find appropriate weights for likelihood calculations

Description

This function takes takes a `matrix` of (possibly binned) data and returns a `matrix` containing the distinct observations, and a `vector` of weights w as described below.

Usage

```
getweights(x)
lcd.getweights(x)
```

Arguments

`x` a data matrix

Details

Given an $n \times d$ matrix x of points in R^d , this function removes duplicated observations, and counts the number of times each observation occurs. This is used to compute a vector w such that

$$w_i = \frac{\# \text{ of times value } i \text{ is observed}}{\# \text{ of observations}}.$$

This function is called by `mlelcd` in order to compute the maximum likelihood estimator when the observed data values are not distinct. In this case, the log likelihood function is of the form

$$\sum_{j=1}^m w_j \log f(X_j),$$

where the sum is over distinct observations.

`lcd.getweights` is deprecated, but retained for compatibility with previous versions.

Value

`xout` A matrix containing the distinct rows of the input matrix `x`
`w` A real-valued vector of weights as described above

Author(s)

Madeleine Cule <mlc40@cam.ac.uk>
 Robert Gramacy
 Richard Samworth

See Also

[mlelcd](#)

Examples

```
## simple normal example

x <- matrix(rnorm(200), ncol=2)
tmp <- getweights(x)
lcd <- mlelcd(tmp$x, tmp$w)
plot(lcd, type="ic")
```

interplcd	<i>Evaluate the log-concave maximum likelihood estimator of 2-d data on a grid for plotting</i>
-----------	---

Description

Evaluates the logarithm of the log-concave maximum likelihood estimator on a grid for 2-d data, for use in `plot.LogConcDEAD`.

Usage

```
interplcd(lcd, gridlen=100 )  
lcd.interp(lcd, gridlen=100 )
```

Arguments

lcd	Object of class "LogConcDEAD" (typically output from <code>mlelcd</code>)
gridlen	A scalar indicating the size of the grid

Details

Interpolates the MLE over a grid.

The output is of a form readily usable by `plot.LogConcDEAD`, `image`, `contour`, etc, as illustrated in the examples below.

For examples, please see `mlelcd`.

`lcd.interp` is deprecated, but retained for compatibility with previous versions.

Value

x	Vector of x -values of the grid
y	Vector of y -values of the grid
z	A <code>matrix</code> of the values of the log of the maximum likelihood estimator at points on the grid

Author(s)

Madeleine Cule <`mle40@cam.ac.uk`>
Robert Gramacy
Richard Samworth

See Also

`mlelcd`

interpmarglcd	<i>Finds marginals of multivariate logconcave maximum likelihood estimators by integrating</i>
---------------	--

Description

Integrates the maximum likelihood estimator of multivariate data over an appropriate subspace to produce axis-aligned marginals for use in `plot.LogConcDEAD`.

Usage

```
interpmarglcd(lcd, marg=1, gridlen=100)
lcd.marg (lcd, marg=1, gridlen=100)
```

Arguments

lcd	Output from <code>mlelcd</code> (of class "LogConcDEAD")
marg	An (integer) scalar indicating which margin is required
gridlen	An (integer) scalar indicating the size of the grid

Details

Given a multivariate log-concave maximum likelihood estimator in the form of an object of class "LogConcDEAD" and a margin `marg`, this function will compute the marginal density estimate $\hat{f}_{n,\text{marg}}$. The estimate is evaluated at `gridlen` equally spaced points in the range where the density estimate is nonzero. These points are given in the vector `x0`.

$\hat{f}_{n,\text{marg}}$ is evaluated by integrating the log-concave maximum likelihood estimator \hat{f}_n over the other components. The marginal density is zero outside the range of `x0`.

For examples, see `mlelcd`.

`lcd.marg` is deprecated, but retained for compatibility with previous versions.

Value

<code>x0</code>	Vector of values at which the marginal density is estimate is computed.
<code>marg</code>	Vector of values of the integrated maximum likelihood estimator at the locations <code>x0</code>

Author(s)

Madeleine Cule <mle40@cam.ac.uk>
 Robert Gramacy
 Richard Samworth

See Also

`dmarglcd`, `mlelcd`

```
mlelcd
```

Compute the maximum likelihood estimator of a log-concave density

Description

Uses Shor's r -algorithm to compute the maximum likelihood estimator of a log-concave density based on an i.i.d. sample. The estimator is uniquely determined by its value at the data points. The output is an object of class "LogConcDEAD" which contains all the information needed to plot the estimator using the `plot` method, or to evaluate it using the function `dlcd`.

Usage

```
mlelcd(x, w=rep(1/nrow(x), nrow(x)), y=initialy(x),
        verbose=-1, alpha=5, c=1, sigmatol=10^-8, integraltol=10^-4,
        ytol=10^-4, stepscale=5.1, stepscale2=2, stepscale3=1.5,
        stepscale4=1.05, desiredsize=3.3, Jtol=0.001, chtol=10^-6)
```

```
lcd.mle(x, w=rep(1/nrow(x), nrow(x)), y=initialy(x),
        verbose=-1, alpha=5, c=1, sigmatol=10^-8, integraltol=10^-4,
        ytol=10^-4, stepscale=5.1, stepscale2=2, stepscale3=1.5,
        stepscale4=1.05, desiredsize=3.3, Jtol=0.001, chtol=10^-6)
```

Arguments

`x` Data in R^d , in the form of an $n \times d$ numeric matrix

`w` Vector of weights w_i such that the computed estimator maximizes

$$\sum_{i=1}^n w_i \log f(x_i)$$

subject to the restriction that f is log-concave. The default is $\frac{1}{n}$ for all i , which corresponds to i.i.d. observations.

`y` Vector giving starting point for the r -algorithm. If none given, a kernel estimate is used.

`verbose`

- -1: (default) prints nothing\
- 0: prints warning messages
- $n > 0$: prints summary information every n iterations

`alpha` Scalar parameter for **SolvOpt**

`c` Scalar giving starting step size

`sigmatol` Real-valued scalar giving one of the stopping criteria: Relative change in σ must be below `sigmatol` for algorithm to terminate. (See Details)

`ytol` Real-valued scalar giving on of the stopping criteria: Relative change in y must be below `ytol` for algorithm to terminate. (See Details)

`integraltol` Real-valued scalar giving one of the stopping criteria: $|1 - \exp(\bar{h}_y)|$ must be below `integraltol` for algorithm to terminate. (See Details)

stepscale, stepscale2, stepscale3, stepscale4, desiredsize	Scalar parameters for SolvOpt . Changing these is not recommended.
Jtol	Parameter controlling when Taylor expansion is used in computing the function σ
chtol	Parameter controlling convex hull computations

Details

The log-concave maximum likelihood density estimator based on data X_1, \dots, X_n is the function that maximizes

$$\sum_{i=1}^n w_i \log f(X_i)$$

subject to the constraint that f is log-concave. For i.i.d.-data, the weights w_i should be $\frac{1}{n}$ for each i .

This is a function of the form \bar{h}_y for some $y \in R^n$, where

$$\bar{h}_y(x) = \inf\{h(x): h \text{ concave}, h(x_i) \geq y_i \text{ for } i = 1, \dots, n\}.$$

Functions of this form may equivalently be specified by dividing C_n , the convex hull of the data, into simplices C_j for $j \in J$ (triangles in 2d, tetrahedra in 3d etc), and setting

$$f(x) = \exp\{b_j^T x - \beta_j\}$$

for $x \in C_j$, and $f(x) = 0$ for $x \notin C_n$.

This function uses Shor's r -algorithm (an iterative subgradient-based procedure) to minimize over vectors y in R^n the function

$$\sigma(y) = -\frac{1}{n} \sum_{i=1}^n y_i + \int \exp(\bar{h}_y(x)) dx.$$

This is equivalent to finding the log-concave maximum likelihood estimator, as demonstrated in *Cule, Samworth and Stewart (2008)*.

An implementation of Shor's r -algorithm based on **SolvOpt** is used.

Computing σ makes use of the **qhull** library, adapted from the R implementation in **geometry**. Code from this package is copied here as it is not currently possible to use compiled code from another package. For points not in general position, this requires a Taylor expansion of σ , discussed in *Cule and D'umbgen (2008)*.

lcd.mle is deprecated, but retained for compatibility with previous versions.

Value

An object of class "LogConcDEAD", with the following components:

x	Data copied from input (may be reordered)
w	weights copied from input (may be reordered)
logMLE	vector of the log of the maximum likelihood estimate, evaluated at the observation points

NumberOfEvaluations	Vector containing the number of steps, number of function evaluations, and number of subgradient evaluations. If the SolvOpt algorithm fails, the first component will be an error code (< 0).
MinSigma	Real-valued scalar giving minimum value of the objective function
b	matrix (see Details)
beta	vector (see Details)
triang	matrix containing final triangulation of the convex hull of the data
verts	matrix containing details of triangulation for use in dlcd
vertsoffset	matrix containing details of triangulation for use in dlcd
chull	Vector containing vertices of faces of the convex hull of the data
outnorm	matrix where each row is an outward pointing normal vectors for the faces of the convex hull of the data. The number of vectors depends on the number of faces of the convex hull.
outoffset	matrix where each row is a point on a face of the convex hull of the data. The number of vectors depends on the number of faces of the convex hull.

Note

For one-dimensional data, the active set algorithm of [logcondens](#) is faster, and may be preferred. The authors gratefully acknowledge the assistance of Lutz Duembgen at the University of Bern for his insight into the objective function σ .

Further references, including definitions and background material, may be found in *Cule, Samworth and Stewart (2008)*.

Author(s)

Madeleine Cule <mlc40@cam.ac.uk>

Robert B. Gramacy

Richard Samworth

References

- Barber, C.B., Dobkin, D.P., and Huhdanpaa, H.T. (1996) *The Quickhull algorithm for convex hulls* ACM Trans. on Mathematical Software, 22(4) p.469-483 <http://www.qhull.org>
- Cule, M. L. and Duembgen, L. (2008) On an auxiliary function for log-density estimation, University of Bern technical report. <http://arxiv.org/abs/0807.4719>
- Cule, M. L., Samworth, R. J., and Stewart, M. I. (2007) *Maximum likelihood estimation of a log-concave density*, Submitted. <http://arxiv.org/abs/0804.3989>
- Kappel, F. and Kuntsevich, A. V. (2000) *An implementation of Shor's r-algorithm* Computational Optimization and Applications 15
<http://www.uni-graz.at/imawww/kuntsevich/solvopt/>
- Shor, N. Z. (1985) *Minimization methods for nondifferentiable functions* Springer-Verlag

See Also

[convhulln](#), [logcondens](#), [interplcd](#), [plot.LogConcDEAD](#), [interpmarglcd](#), [rlcd](#), [dlcd](#), [dmarglcd](#)

Examples

```
## Some simple normal data, and a few plots

x <- matrix(rnorm(200), ncol=2)
lcd <- mlelcd(x)
g <- interplcd(lcd)
par(mfrow=c(2,2), ask=TRUE)
plot(lcd, g=g, type="c")
plot(lcd, g=g, type="c", uselog=TRUE)
plot(lcd, g=g, type="i")
plot(lcd, g=g, type="i", uselog=TRUE)

## Some plots of marginal estimates
par(mfrow=c(1,1))
g.marg1 <- interpmarglcd(lcd, marg=1)
g.marg2 <- interpmarglcd(lcd, marg=2)
plot(lcd, marg=1, g.marg=g.marg1)
plot(lcd, marg=2, g.marg=g.marg2)

## generate some points from the fitted density
generated <- rlcd(100, lcd)
genmean <- mean(generated)

## evaluate the fitted density
mypoint <- c(0, 0)
dlcd(mypoint, lcd, uselog=FALSE)
mypoint <- c(10, 0)
dlcd(mypoint, lcd, uselog=FALSE)

## evaluate the marginal density
dmarglcd(0, lcd, marg=1)
dmarglcd(1, lcd, marg=2)
```

plot.LogConcDEAD *Plot a log-concave maximum likelihood estimator*

Description

[plot](#) method for class "LogConcDEAD". Plots of various types are available for 1- and 2-d data. For dimension greater than 1, plots of axis-aligned marginal density estimates are available.

Usage

```
## S3 method for class 'LogConcDEAD':
plot(x, uselog=FALSE, type="ic", addp=TRUE,
     drawlabels=TRUE, gridlen=100, g, marg, g.marg, main, xlab, ylab, ...)
```

Arguments

<code>x</code>	Object of class "LogConcDEAD" (typically output from <code>mlelcd</code>)
<code>uselog</code>	Scalar logical: should the plot be on the log scale?
<code>type</code>	Plot type: "p" perspective, "c" contour, "i" image, <code>ic</code> image and contour, <code>r</code> using <code>rgl</code> (the best!)
<code>addp</code>	Scalar logical: should the data points be plotted? (On the surface for $d \geq 2$; as tick marks for $d = 1$)
<code>drawlabels</code>	Scalar logical: should labels be added to contour lines? (only relevant for types "ic" and "c")
<code>gridlen</code>	Integer scalar indicating the number of points at which the maximum likelihood estimator is evaluated in each dimension
<code>g</code>	(optional) a matrix of density estimate values (the result of a call to <code>interplcd</code>). If many plots of a single dataset are required, it may be quicker to compute the grid using <code>interplcd(x)</code> and pass the result to <code>plot</code>
<code>marg</code>	If non-NULL, this scalar integer determines which marginal should be plotted (should be between 1 and d)
<code>g.marg</code>	If <code>g</code> is non-NULL, can contain a vector of marginal density estimate values (the output of <code>interp marglcd</code>). If many plots of a single dataset are required, it may be quicker to compute the marginal values to compute marginal values using <code>interp marglcd</code> and pass the result to <code>plot</code>
<code>main</code>	Title
<code>xlab</code>	x-axis label
<code>ylab</code>	y-axis label
<code>...</code>	Other arguments to be passed to the generic <code>plot</code> method

Details

The density estimate is evaluated on a grid of points using the `interplcd` function. If several plots are required, this may be computed separately and passed to `plot` using the `g` argument.

For two dimensional data, the default plot type is "ic", corresponding to `image` and `contour` plots. These may be obtained separately using plot type "i" or "c" respectively. Where available, the use of plot type "r" is recommended. This uses the `rgl` package to produce a 3-d plot that may be rotated by the user. The option "p" produces perspective plots.

For data of dimension at least 2, axis-aligned marginals may be plotted by setting the `marg` argument. This integrates the estimated density over the remaining dimensions. If several plots are required, the estimate may be computed using the function `interp marglcd` and passed using the argument `g.marg`.

Where relevant, the colors were obtained from the function `heat_hcl` in the package **colorspace**. Thanks to Achim Zeileis for this suggestion.

For examples, see `mlelcd`.

Author(s)

Madeleine Cule <mle40@cam.ac.uk>

Robert B. Gramacy

Richard Samworth

See Also

`mlelcd`, `interplcd`, `interpmarglcd`, `heat_hcl`

`print.LogConcDEAD` *Summarizing log-concave maximum likelihood estimator*

Description

Generic `print` and `summary` method for objects of class `"LogConcDEAD"`

Usage

```
## S3 method for class 'LogConcDEAD':
print(x, ...)
## S3 method for class 'LogConcDEAD':
summary(object, ...)
```

Arguments

<code>x</code>	Object of class <code>"LogConcDEAD"</code> (typically output from <code>mlelcd</code>), as required by <code>print</code>
<code>object</code>	Object of class <code>"LogConcDEAD"</code> (typically output from <code>mlelcd</code>), as required by <code>summary</code>
<code>...</code>	Other arguments passed to <code>print</code> or <code>summary</code>

Details

`print` and `summary` currently perform the same function.

If there has been an error computing the maximum likelihood estimator, an error message is printed.

Otherwise, the value of the log maximum likelihood estimator at observation points is printed. The number of iterations required by the subgradient and the number of function evaluations are also printed.

Author(s)

Madeleine Cule <mlc40@cam.ac.uk>

Robert B. Gramacy

Richard Samworth

See Also

[mle1cd](#)

r1cd

Sample from a log-concave maximum likelihood estimate

Description

Draws independent samples a log-concave maximum likelihood estimate. The estimate should be specified in the form of an object of class "LogConcDEAD", the result of a call to [mle1cd](#).

Usage

```
r1cd(n, lcd)
lcd.sample(lcd, nsample=1)
```

Arguments

`n`, `nsample` A scalar integer indicating the number of samples required

`lcd` Object of class "LogConcDEAD" (typically output from [mle1cd](#))

Details

This function uses a simple rejection sampling scheme to draw independent random samples from a log-concave maximum likelihood estimator.

For examples, see [mle1cd](#).

`lcd.sample` is deprecated, but retained for compatibility with previous versions of this package.

Value

A numeric matrix with `nsample` rows, each row corresponding to a point in R^d drawn from the distribution with density defined by `lcd`.

Note

Details of the rejection sampling scheme can be found in *Cule, Samworth and Stewart (2008)*

Author(s)

Madeleine Cule <mlc40@cam.ac.uk>

Robert Gramacy

Richard Samworth

References

Cule, M. L., Samworth, R. J. and Stewart, M. I. (2007) *Maximum likelihood estimation of a log-concave density* Submitted, available from arXiv at <http://arxiv.org/abs/0804.3989>

See Also

[mlelcd](#)

Index

- *Topic **datagen**
 - rlcd, 17
- *Topic **distribution**
 - dlcd, 3
 - rlcd, 17
- *Topic **dplot**
 - dmarglcd, 5
 - interplcd, 9
 - interpmarglcd, 10
 - plot.LogConcDEAD, 14
- *Topic **dynamic**
 - plot.LogConcDEAD, 14
- *Topic **hplot**
 - plot.LogConcDEAD, 14
- *Topic **iplot**
 - plot.LogConcDEAD, 14
- *Topic **multivariate**
 - dlcd, 3
 - dmarglcd, 5
 - getinfoLCD, 6
 - getweights, 7
 - interplcd, 9
 - interpmarglcd, 10
 - LogConcDEAD-package, 1
 - mleLCD, 11
 - plot.LogConcDEAD, 14
 - print.LogConcDEAD, 16
 - rlcd, 17
- *Topic **nonparametric**
 - dlcd, 3
 - dmarglcd, 5
 - getinfoLCD, 6
 - getweights, 7
 - interplcd, 9
 - interpmarglcd, 10
 - LogConcDEAD-package, 1
 - mleLCD, 11
 - plot.LogConcDEAD, 14
 - print.LogConcDEAD, 16
- rlcd, 17
- *Topic **package**
 - LogConcDEAD-package, 1
- contour, 9, 15
- convhulln, 14
- dlcd, 2, 3, 7, 11, 13, 14
- dmarglcd, 2, 5, 10, 14
- getinfoLCD, 6
- getweights, 7
- heat_hcl, 16
- image, 9, 15
- interplcd, 2, 9, 14–16
- interpmarglcd, 2, 10, 14–16
- lcd.eval (dlcd), 3
- lcd.getweights (getweights), 7
- lcd.interp (interplcd), 9
- lcd.marg (interpmarglcd), 10
- lcd.marg.eval (dmarglcd), 5
- lcd.mle (mleLCD), 11
- lcd.sample (rlcd), 17
- LogConcDEAD
 - (LogConcDEAD-package), 1
- LogConcDEAD-package, 1
- logcondens, 2, 13, 14
- mleLCD, 2, 4–10, 11, 15–18
- plot, 11, 14, 15
- plot.LogConcDEAD, 2, 9, 10, 14, 14
- print, 2
- print.LogConcDEAD, 16
- rgl, 2, 15
- rlcd, 2, 14, 17
- summary, 2

`summary.LogConcDEAD`
`(print.LogConcDEAD)`, [16](#)