

# Package ‘LogConcDEAD’

March 13, 2025

**Type** Package

**Title** Log-Concave Density Estimation in Arbitrary Dimensions

**Version** 1.6-12

**Date** 2025-03-13

**Depends** R (>= 3.0)

**Imports** MASS, mclust, mvtnorm

**Suggests** rgl, tkrplot

**Enhances** logcondens

**Description** Software for computing a log-concave (maximum likelihood) estimator for independent and identically distributed data in any number of dimensions. For a detailed description of the method see Cule, Samworth and Stewart (2010, Journal of Royal Statistical Society Series B, <[doi:10.1111/j.1467-9868.2010.00753.x](https://doi.org/10.1111/j.1467-9868.2010.00753.x)>).

**License** GPL (>= 2)

**NeedsCompilation** yes

**Author** Madeleine Cule [aut],  
Robert Gramacy [aut],  
Richard Samworth [aut],  
Yining Chen [aut, cre],  
Lutz Duembgen [ctb] (helped improve the numerical stability),  
Vikneswaran Gopal [ctb] (proposed the Metropolis-Hastings scheme for draw samples from a log-concave distribution),  
George Casella [ctb] (proposed the Metropolis-Hastings scheme for draw samples from a log-concave distribution),  
C. Bradford Barber [cph] (author of the Qhull library),  
The Geometry Center [cph],  
Kai Habel [cph] (author of the function that imports convex hull computation from Qhull to R),  
Raoul Grasman [cph] (author of the function that imports convex hull computation from Qhull to R),  
Alexei Kuntsevich [cph] (author of the C function for Shor's r algorithm),  
Franz Kappel [cph] (author of the C function for Shor's r algorithm)

**Maintainer** Yining Chen <y.chen101@lse.ac.uk>

**Repository** CRAN

**Date/Publication** 2025-03-13 12:00:06 UTC

## Contents

LogConcDEAD-package . . . . .	2
cov.LogConcDEAD . . . . .	4
dlcd . . . . .	5
dmarglcd . . . . .	6
dslcd . . . . .	7
EMmixlcd . . . . .	8
getinfolcd . . . . .	10
getweights . . . . .	12
hatA . . . . .	13
interactive2D . . . . .	14
interplcd . . . . .	15
interpmarglcd . . . . .	16
mlelcd . . . . .	17
plot.LogConcDEAD . . . . .	21
print.LogConcDEAD . . . . .	23
rlcd . . . . .	24
rslcd . . . . .	25
<b>Index</b>	<b>27</b>

---

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

---

## 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.

`hatA` calculates the difference between the sample covariance and the fitted covariance.

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

`dslcd` evaluates the smoothed version of estimated density at a particular point.

`rlcd` draws samples from the estimated density.  
`rslcd` draws samples from the smoothed version of 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)

Yining Chen (maintainer) <y.chen101@lse.ac.uk>

Madeleine Cule

Robert 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>
- Chen, Y. and Samworth, R. J. (2013) *Smoothed log-concave maximum likelihood estimation with applications* Statist. Sinica, 23, 1373-1398. <https://arxiv.org/abs/1102.1191v4>
- Cule, M. L. and Duembgen, L. (2008) *On an auxiliary function for log-density estimation*, University of Bern technical report. <https://arxiv.org/abs/0807.4719>
- Cule, M. L., Samworth, R. J., and Stewart, M. I. (2010) *Maximum likelihood estimation of a multi-dimensional log-concave density* J. Roy. Statist. Soc., Ser. B. (with discussion), 72, 545-600.
- Gopal, V. and Casella, G. (2010) *Discussion of Maximum likelihood estimation of a log-concave density by Cule, Samworth and Stewart* J. Roy. Statist. Soc., Ser. B., 72, 580-582.
- Grundmann, A. and Moeller, M. (1978) *Invariant Integration Formulas for the N-Simplex by Combinatorial Methods* SIAM Journal on Numerical Analysis, Volume 15, Number 2, 282-290.
- Kappel, F. and Kuntsevich, A. V. (2000) *An implementation of Shor's r-algorithm* Computational Optimization and Applications, Volume 15, Issue 2, 193-205.
- 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)
oldpar <- par(mfrow = c(1,1))
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)
par(oldpar)

## Some plots of marginal estimates
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 <- colMeans(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)

```

---

cov.LogConcDEAD

*Compute the covariance matrix of a log-concave maximum likelihood estimator*

---

**Description**

This function computes the covariance matrix of a log-concave maximum likelihood estimator.

**Usage**

```
cov.LogConcDEAD(lcd)
```

**Arguments**

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

**Details**

This function evaluates the covariance matrix of a given log-concave maximum likelihood estimator using the second order partial derivatives of the auxiliary function studied in *Cule, M. L. and Duembgen, L. (2008)*.

For examples, see [mlelcd](#).

**Value**

A matrix equals the covariance matrix of the log-concave maximum likelihood density estimator.

**Author(s)**

Yining Chen

Madeleine Cule

Robert Gramacy

Richard Samworth

**References**

Cule, M. L. and Duembgen, L. (2008) On an auxiliary function for log-density estimation, University of Bern technical report. <https://arxiv.org/abs/0807.4719>

**See Also**

[hata](#)

---

 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)
```

**Arguments**

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

**Details**

A log-concave maximum likelihood estimate  $\hat{f}_n$  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](#).

**Value**

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

**Author(s)**

Madeleine Cule  
Robert Gramacy  
Richard Samworth

**See Also**

[mlelcd](#)

---

dmarglcd

*Evaluate the marginal of multivariate log-concave maximum likelihood estimators at a point*

---

**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)

**Arguments**

x	Point (or vector of points) at which the marginal density is to be evaluated
lcd	Object of class "LogConcDEAD" (typically output from <a href="#">mlelcd</a> )
marg	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](#).

**Value**

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

**Author(s)**

Madeleine Cule  
Robert Gramacy  
Richard Samworth

**See Also**

[mlelcd](#)

---

dslcd	<i>Evaluation of a smoothed log-concave maximum likelihood estimator at given points</i>
-------	--

---

**Description**

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

**Usage**

```
dslcd(x, lcd, A=hatA(lcd))
```

**Arguments**

x	Point (or matrix of points) at which the smoothed log-concave maximum likelihood estimator should be evaluated
lcd	Object of class "LogConcDEAD" (typically output from <a href="#">mlelcd</a> )
A	A positive definite matrix that determines the degree of smoothing, typically taken as the output of <code>hatA(lcd)</code>

**Details**

The smoothed log-concave maximum likelihood estimator is a fully automatic nonparametric density estimator, obtained as a canonical smoothing of the log-concave maximum likelihood estimator. More precisely, it equals the convolution  $\hat{f} * \phi_{d, \hat{A}}$ , where  $\phi_{d, \hat{A}}$  is the density function of  $d$ -dimensional multivariate normal with covariance matrix  $\hat{A}$ . Typically,  $\hat{A}$  is taken as the difference between the sample covariance and the covariance of fitted log-concave maximum likelihood density. Therefore, this estimator matches both the empirical mean and empirical covariance.

The estimate is evaluated numerically either by Gaussian quadrature in two dimensions, or in higher dimensions, via a combinatorial method proposed by *Grundmann and Moeller (1978)*. Details of the computational aspects can be found in *Chen and Samworth (2011)*. In one dimension, explicit expression can be derived. See [logcondens](#) for more information.

For examples, see [mlelcd](#)

**Value**

A vector of smoothed log-concave maximum likelihood estimate values, as evaluated at the points  $x$ .

**Author(s)**

Yining Chen

Madeleine Cule

Robert Gramacy

Richard Samworth

**References**

Chen, Y. and Samworth, R. J. (2013) *Smoothed log-concave maximum likelihood estimation with applications* Statist. Sinica, 23, 1373-1398. <https://arxiv.org/abs/1102.1191v4>

Grundmann, A. and Moeller, M. (1978) *Invariant Integration Formulas for the N-Simplex by Combinatorial Methods* SIAM Journal on Numerical Analysis, Volume 15, Number 2, 282-290.

**See Also**

[dlcd](#), [hatA](#), [mlelcd](#)

---

EMmixlcd

*Estimate the mixture proportions and component densities using EM algorithm*

---

**Description**

Uses EM algorithm to estimate the mixture proportions and the component densities. The output is an object of class "lcmix" which contains mixture proportions at each observation and all the information of the estimated component densities.

**Usage**

```
EMmixlcd( x, k = 2, y, props, epsratio=10^-6, max.iter=50,
          epstheta=10^-8, verbose=-1 )
```

**Arguments**

x	Data in $R^d$ , in the form of an $n \times d$ numeric matrix
k	The number of components, equals 2 by default
y	An $n \times k$ numeric matrix giving the starting values for the EM algorithm. If none given, a hierarchical Gaussian clustering model is used. To reduce the computational burden while allowing sufficient flexibility for the EM algorithm, it is recommended to leave this argument unspecified.
props	Vector of length $k$ containing the starting value of proportions. If none given, a hierarchical Gaussian clustering model is used. To reduce the computational burden while allowing sufficient flexibility for the EM algorithm, it is recommended to leave this argument unspecified.
epsratio	EM algorithm will terminate if the increase in the proportion of the likelihood is less than this specified ratio. Default value is $10^{-6}$ .
max.iter	The maximum number of iterations for the EM algorithm
epstheta	$epstheta/n$ is the threshold of the weight below which data point is discarded from the cluster. This quantity is introduced to increase the computational efficiency and stability.
verbose	<ul style="list-style-type: none"> <li>• -1: (default) prints nothing</li> <li>• 0: prints warning messages</li> <li>• &gt; 0: prints summary information every <math>n</math> iterations</li> </ul>

**Details**

An introduction to the Em algorithm can be found in *McLachlan and Krishnan (1997)*. Briefly, given the current estimates of the mixture proportions and component densities, we first update the estimates of the mixture proportions. We then update the estimates of the component densities by using `mlelcd`. In fact, the incorporation of the weights in the maximization process in `mlelcd` presents no additional complication.

In our case, because of the computational intensity of the method, we first cluster the points according to a hierarchical Gaussian clustering model and then iterate the EM algorithm until the increase in the proportion of the likelihood is less than a pre-specified quantity at each step.

More technical details can be found in *Cule, Samworth and Stewart(2010)*

**Value**

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

x	Data copied from input (may be reordered)
logf	An $n \times k$ matrix of the log of the maximum likelihood estimate, evaluated at the observation points for each component.

props            Vector containing the estimated proportions of components  
 niter            Number of iterations of the EM algorithm  
 lcdloglik        The log-likelihood after the final iteration

### Author(s)

Yining Chen  
 Madeleine Cule  
 Robert B. Gramacy  
 Richard Samworth

### References

Cule, M. L., Samworth, R. J., and Stewart, M. I. (2010) *Maximum likelihood estimation of a log-concave density*, Journal of the Royal Statistical Society, Series B, 72(5) p.545-607.  
 McLachlan, G. J. and Krishnan, T. (1997) *The EM Algorithm and Extensions*, New York: Wiley.

### See Also

[mclust](#), [logcondens](#), [plot.LogConcDEAD](#), [mleLCD](#), [dlcd](#)

### Examples

```
##Simple bivariate normal data
set.seed( 1 )
n = 15
d = 2
props=c( 0.6, 0.4 )
shift=2
x <- matrix( rnorm( n*d ), ncol = d )
shiftvec <- ifelse( runif( n ) > props[ 1 ], 0, shift )
x[,1] <- x[,1] + shiftvec
EMmixLCD( x, k = 2, max.iter = 2)
```

---

getinfoLCD

*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

```
getinfoLCD(x, y, w = rep(1/length(y), length(y)), chto1 = 10^-6,
  MinSigma = NA, NumberOfEvaluations = NA)
```

**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 <b>SolvOpt</b> 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 <b>SolvOpt</b> algorithm fails, the first component will be an error code ( $< 0$ ).
MinSigma	Real-valued scalar giving minimum value of the objective function
b	matrix containing row by row the values of $b_j$ 's corresponding to each triangulation; see also <a href="#">mleLCD</a>
beta	vector containing the values of $\beta_j$ 's corresponding to each triangulation; see also <a href="#">mleLCD</a>
triang	matrix containing final triangulation of the convex hull of the data
verts	matrix containing details of triangulation for use in <a href="#">dlcd</a>
vertsoffset	matrix containing details of triangulation for use in <a href="#">dlcd</a>
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  
 Robert B. Gramacy  
 Richard Samworth  
 Yining Chen

**See Also**

[mlelcd](#)

---

getweights

*Find appropriate weights for likelihood calculations*

---

**Description**

This function 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)
```

**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.

**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  
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")
```

---

hatA	<i>Compute the smoothing matrix of the smoothed log-concave maximum likelihood estimator</i>
------	--

---

**Description**

This function computes the matrix  $\hat{A}$  of the smoothed log-concave maximum likelihood estimator

**Usage**

```
hatA(lcd)
```

**Arguments**

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

**Details**

This function evaluates the the matrix  $\hat{A}$  of the smoothed log-concave maximum likelihood estimator, which is positive definite, and equals the difference between the sample covariance matrix and the covariance matrix of the fitted log-concave maximum likelihood density estimator.

For examples, see [mlelcd](#)

**Value**

A matrix equals  $\hat{A}$  of the smoothed log-concave maximum likelihood estimator

**Note**

Details of the computational aspects can be found in *Chen and Samworth (2011)*.

**Author(s)**

Yining Chen  
Madeleine Cule  
Robert Gramacy  
Richard Samworth

**References**

Chen, Y. and Samworth, R. J. (2013) *Smoothed log-concave maximum likelihood estimation with applications* Statist. Sinica, 23, 1373-1398. <https://arxiv.org/abs/1102.1191v4>

**See Also**

[cov.LogConcDEAD](#)

---

interactive2D	<i>A GUI for classification in two dimensions using smoothed log-concave</i>
---------------	--

---

**Description**

Uses [tkrplot](#) to create a GUI for two-class classification in two dimensions using the smoothed log-concave maximum likelihood estimates

**Usage**

```
interactive2D(data, cl)
```

**Arguments**

data	Data in $R^2$ , in the form of an $n \times 2$ numeric matrix
cl	factor of true classifications of the data set

**Details**

This function uses [tkrplot](#) to create a GUI for two-class classification in two dimensions using the smoothed log-concave maximum likelihood estimates. The construction of the classifier is standard, and can be found in *Chen and Samworth (2013)*. The slider controls the risk ratio of two classes (equals one by default), which provides a way of demonstrating how the decision boundaries change as the ratio varies. Observations from different classes are plotted in red and green respectively.

**Value**

A GUI with a slider

**Author(s)**

Yining Chen  
Madeleine Cule  
Robert B. Gramacy  
Richard Samworth

**References**

Chen, Y. and Samworth, R. J. (2013) *Smoothed log-concave maximum likelihood estimation with applications* Statist. Sinica, 23, 1373-1398. <https://arxiv.org/abs/1102.1191v4>

Cule, M. L., Samworth, R. J., and Stewart, M. I. (2010) *Maximum likelihood estimation of a log-concave density*, Journal of the Royal Statistical Society, Series B, 72(5) p.545-607.

**See Also**

[dslcd,mlelcd](#)

**Examples**

```
## Simple bivariate normal data
## only works interactively, not run as a test example here
if(interactive()){
  set.seed( 1 )
  n = 15
  d = 2
  props=c( 0.6, 0.4 )
  x <- matrix( rnorm( n*d ), ncol = d )
  shiftvec <- ifelse( runif( n ) > props[ 1 ], 0, 1)
  x[,1] <- x[,1] + shiftvec
  interactive2D( x, shiftvec )
}
```

---

interplcd

*Evaluate the log-concave maximum likelihood estimator of 2-d data on a grid for plotting*

---

**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 )
```

**Arguments**

lcd	Object of class "LogConcDEAD" (typically output from <a href="#">mlelcd</a> )
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](#).

**Value**

x	Vector of $x$ -values of the grid
y	Vector of $y$ -values of the grid
z	A matrix of the values of the log of the maximum likelihood estimator at points on the grid

**Author(s)**

Madeleine Cule  
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)
```

**Arguments**

lcd	Output from <a href="#">mlelcd</a> (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 `xo`.

$\hat{f}_{n,\text{marg}}$  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 `xo`.

For examples, see [mlelcd](#).

**Value**

<code>xo</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>xo</code>

**Author(s)**

Madeleine Cule  
Robert Gramacy  
Richard Samworth

**See Also**

[dmarglcd](#), [mlelcd](#)

---

mlelcd	<i>Compute the maximum likelihood estimator of a log-concave density</i>
--------	--

---

**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, 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	<ul style="list-style-type: none"> <li>• -1: (default) prints nothing</li> <li>• 0: prints warning messages</li> <li>• <math>n &gt; 0</math>: prints summary information every <math>n</math> iterations</li> </ul>
alpha	Scalar parameter for <b>SolvOpt</b>
c	Scalar giving starting step size
sigmatol	Real-valued scalar giving one of the stopping criteria: Relative change in $\sigma$ must be below sigmatol for algorithm to terminate. (See Details)
ytol	Real-valued scalar giving one 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)
Jtol	Parameter controlling when Taylor expansion is used in computing the function $\sigma$
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  $\sigma$  makes use of the **qhull** library. Code from this C-based library is copied here as it is not currently possible to use compiled code from another library. For points not in general position, this requires a Taylor expansion of  $\sigma$ , discussed in *Cule and Duembgen (2008)*.

### 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 <b>SolvOpt</b> algorithm fails, the first component will be an error code ( $< 0$ ).
MinSigma	Real-valued scalar giving minimum value of the objective function
b	matrix containing row by row the values of $b_j$ 's corresponding to each triangulation; see also the Details section above
beta	vector containing the values of $\beta_j$ 's corresponding to each triangulation; see also the Details section above
triang	matrix containing final triangulation of the convex hull of the data
verts	matrix containing details of triangulation for use in <b>dlcd</b>
vertsoffset	matrix containing details of triangulation for use in <b>dlcd</b>
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  $\sigma$ .

Further references, including definitions and background material, may be found in *Cule, Samworth and Stewart (2010)*.

**Author(s)**

Madeleine Cule  
 Robert B. Gramacy  
 Richard Samworth  
 Yining Chen

**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. <https://arxiv.org/abs/0807.4719>
- Cule, M. L., Samworth, R. J., and Stewart, M. I. (2010) *Maximum likelihood estimation of a log-concave density*, Journal of the Royal Statistical Society, Series B, 72(5) p.545-607.
- Kappel, F. and Kuntsevich, A. V. (2000) *An implementation of Shor's r-algorithm*, Computational Optimization and Applications, Volume 15, Issue 2, 193-205.
- Shor, N. Z. (1985) *Minimization methods for nondifferentiable functions*, Springer-Verlag

**See Also**

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

**Examples**

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

x <- matrix(rnorm(200),ncol=2)
lcd <- mlelcd(x)
g <- interplcd(lcd)

oldpar <- par(mfrow = c(1,1))
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)
par(oldpar)

## 2D interactive plot (need rgl package, not run here)
if(interactive()) {plot(lcd, type="r")}
```

```
## Some plots of marginal estimates
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
## via independent rejection sampling
generated1 <- rlcd(100, lcd)
colMeans(generated1)
## via Metropolis-Hastings algorithm
generated2 <- rlcd(100, lcd, "MH")
colMeans(generated2)

## evaluate the fitted density
mypoint <- c(0, 0)
dlcd(mypoint, lcd, uselog=FALSE)
mypoint <- c(1, 0)
dlcd(mypoint, lcd, uselog=FALSE)

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

## evaluate the covariance matrix of the fitted density
covariance <- cov.LogConcDEAD(lcd)

## find the hat matrix for the smoothed log-concave that
## matches empirical mean and covariance
A <- hatA(lcd)

## evaluate the fitted smoothed log-concave density
mypoint <- c(0, 0)
dslcd(mypoint, lcd, A)
mypoint <- c(1, 0)
dslcd(mypoint, lcd, A)

## generate some points from the fitted smoothed log-concave density
generated <- rslcd(100, lcd, A)

```

---

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=400, g, marg, g.marg, main, xlab, ylab, ...)

```

**Arguments**

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

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

For examples, see [mlelcd](#).

**Value**

No return value, plot will display

**Author(s)**

Madeleine Cule  
Robert B. Gramacy  
Richard Samworth  
Yining Chen

**See Also**

[mlelcd](#), [interplcd](#), [interp marglcd](#), [heat\\_hcl](#)

---

print.LogConcDEAD	<i>Summarizing log-concave maximum likelihood estimator</i>
-------------------	---

---

**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**

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

**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.

**Value**

No return value, log MLE at observation points will be printed out on the screen.

**Author(s)**

Madeleine Cule  
 Robert B. Gramacy  
 Richard Samworth

**See Also**

[mle1cd](#)

---

r1cd

*Sample from a log-concave maximum likelihood estimate*

---

**Description**

Draws samples from 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=1, lcd, method=c("Independent", "MH"))
```

**Arguments**

n	A scalar integer indicating the number of samples required
lcd	Object of class "LogConcDEAD" (typically output from <a href="#">mle1cd</a> )
method	Indicator of the method used to draw samples, either via independent rejection sampling (default choice) or via Metropolis-Hastings

**Details**

This function by default uses a simple rejection sampling scheme to draw independent random samples from a log-concave maximum likelihood estimator. One can also use the Metropolis-Hastings option to draw (dependent) samples with a higher acceptance rate.

For examples, see [mle1cd](#).

**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 can be found in Appendix B.3 of *Cule, Samworth and Stewart (2010)*. Details of the Metropolis-Hastings scheme can be found in *Gopal and Casella (2010)*

**Author(s)**

Yining Chen  
 Madeleine Cule  
 Robert Gramacy  
 Richard Samworth

**References**

Cule, M. L., Samworth, R. J., and Stewart, M. I. (2010) *Maximum likelihood estimation of a multi-dimensional log-concave density* J. Roy. Statist. Soc., Ser. B. (with discussion), 72, 545-600.

Gopal, V. and Casella, G. (2010) *Discussion of Maximum likelihood estimation of a log-concave density by Cule, Samworth and Stewart* J. Roy. Statist. Soc., Ser. B., 72, 580-582.

**See Also**

[mlelcd](#)

---

rslcd	<i>Sample from a smoothed log-concave maximum likelihood estimate</i>
-------	---

---

**Description**

Draws samples from a smoothed 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 [mlelcd](#), and a positive definite matrix.

**Usage**

```
rslcd(n=1, lcd, A=hatA(lcd), method=c("Independent", "MH"))
```

**Arguments**

n	A scalar integer indicating the number of samples required
lcd	Object of class "LogConcDEAD" (typically output from <a href="#">mlelcd</a> )
A	A positive definite matrix that determines the degree of smoothing, typically taken as the output of <code>hatA(lcd)</code>
method	Indicator of the method used to draw samples, either via independent rejection sampling (default choice) or via Metropolis-Hastings

**Details**

This function by default uses a simple rejection sampling scheme to draw independent random samples from a smoothed log-concave maximum likelihood estimator. One can also use the Metropolis-Hastings option to draw (dependent) samples with a higher acceptance rate.

For examples, see [mlelcd](#).

**Value**

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

**Author(s)**

Yining Chen

Madeleine Cule

Robert Gramacy

Richard Samworth

**References**

Chen, Y. and Samworth, R. J. (2013) *Smoothed log-concave maximum likelihood estimation with applications* Statist. Sinica, 23, 1373-1398. <https://arxiv.org/abs/1102.1191v4>

Cule, M. L., Samworth, R. J., and Stewart, M. I. (2010) *Maximum likelihood estimation of a multi-dimensional log-concave density* J. Roy. Statist. Soc., Ser. B. (with discussion), 72, 545-600.

Gopal, V. and Casella, G. (2010) *Discussion of Maximum likelihood estimation of a log-concave density by Cule, Samworth and Stewart* J. Roy. Statist. Soc., Ser. B., 72, 580-582.

**See Also**

[mlclcd](#), [rlcd](#), [hata](#)

# Index

- \* **EM**
  - EMmixlcd, 8
- \* **classification**
  - EMmixlcd, 8
  - interactive2D, 14
- \* **datagen**
  - rlcd, 24
  - rslcd, 25
- \* **distribution**
  - dlcd, 5
  - rlcd, 24
  - rslcd, 25
- \* **dplot**
  - dmarglcd, 6
  - interplcd, 15
  - interpmarglcd, 16
  - plot.LogConcDEAD, 21
- \* **dynamic**
  - plot.LogConcDEAD, 21
- \* **hplot**
  - plot.LogConcDEAD, 21
- \* **iplot**
  - plot.LogConcDEAD, 21
- \* **multivariate**
  - cov.LogConcDEAD, 4
  - dlcd, 5
  - dmarglcd, 6
  - dslcd, 7
  - EMmixlcd, 8
  - getinfoLCD, 10
  - getweights, 12
  - hatA, 13
  - interactive2D, 14
  - interplcd, 15
  - interpmarglcd, 16
  - LogConcDEAD-package, 2
  - mlelcd, 17
  - plot.LogConcDEAD, 21
  - print.LogConcDEAD, 23
  - rlcd, 24
  - rslcd, 25
- \* **nonparametric**
  - cov.LogConcDEAD, 4
  - dlcd, 5
  - dmarglcd, 6
  - dslcd, 7
  - EMmixlcd, 8
  - getinfoLCD, 10
  - getweights, 12
  - hatA, 13
  - interactive2D, 14
  - interplcd, 15
  - interpmarglcd, 16
  - LogConcDEAD-package, 2
  - mlelcd, 17
  - plot.LogConcDEAD, 21
  - print.LogConcDEAD, 23
  - rlcd, 24
  - rslcd, 25
- \* **package**
  - LogConcDEAD-package, 2
- \* **smoothing**
  - dslcd, 7
  - hatA, 13
  - LogConcDEAD-package, 2
  - rslcd, 25
- contour, 16, 22
- cov.LogConcDEAD, 4, 14, 20
- dlcd, 2, 5, 8, 10, 11, 17, 19, 20
- dmarglcd, 3, 6, 17, 20
- dslcd, 2, 7, 15
- EMmixlcd, 8
- getinfoLCD, 10
- getweights, 12
- hatA, 2, 5, 8, 13, 26

heat\_hcl, 23

image, 16, 22

interactive2D, 14

interplcd, 3, 15, 20, 22, 23

interp marglcd, 3, 16, 20, 22, 23

LogConcDEAD (LogConcDEAD-package), 2

LogConcDEAD-package, 2

logcondens, 3, 8, 10, 19, 20

mclust, 10

mlelcd, 2–13, 15–17, 17, 22–26

plot, 17, 21, 22

plot.LogConcDEAD, 3, 10, 15, 16, 20, 21

print, 3

print.LogConcDEAD, 23

rgl, 3, 22

rlcd, 3, 20, 24, 26

rslcd, 3, 25

summary, 3

summary.LogConcDEAD  
(print.LogConcDEAD), 23

tkrplot, 14