Package 'cnmlcd'

February 16, 2018

Title Maximum Likelihood Estimation of a Log-Concave Density Function

Version 1.2-0

Date 2018-02-16

Author Yu Liu, Yong Wang

Maintainer Yong Wang <yongwang@auckland.ac.nz>

Depends lsei

Description

Contains functions for computing the nonparametric maximum likelihood estimate of a log-concave density function from univariate observations. This log-density estimate is always a piecewise linear function.

License GPL (>= 2)

LazyData TRUE

NeedsCompilation no

Repository CRAN

Date/Publication 2018-02-16 04:29:40 UTC

R topics documented:

cnmlcd *Maximum Likelihood Estimation of a Log-concave Density Function*

Description

The function computes the nonparametric maximum likelihood estimate (NPMLE) of a log-concave density from univariate observations.

Usage

```
cnmlcd(x, lcd, maxit = 100, tol = 1e-06,
       plot=c("null","density","logdensity","gradient"))
```

Arguments

x	a vector storing univariate observations.
lcd	an initial log-concave density function, which is an object of class "lcd".
maxit	maximum number of iterations.
tol	tolerance level for stopping the algorithm. Internally, it is used as the threshold on the increase of the log-likelihood after each iteration.
plot	type of a plot to be created in each iteration of the algorithm. If "null", no plot is produced. The argument is mainly for monitoring purposes.

Details

The algorithm used to compute the NPMLE is an extension of the constrained Newton method of Wang (2007) for nonparametric mixture estimation. It guarantees to find the unique NPMLE.

The algorithm is described in Liu and Wang (2018).

Value

lcd	the NPMLE when the algorithm successfully converges, stored as an object of class "lcd".
ll	the log-likelihood value, evaluated at the NPMLE.
num.iterations	number of iterations used.
max.gradient	maximum gradient value at convergence.
convergence	= 0, a successful convergence; = 1, failure of convergence, likely because the number of iterations is reached.

Author(s)

Yu Liu <liu.yu@auckland.ac.nz>, Yong Wang <yongwang@auckland.ac.nz>

References

Wang, Y. (2007). On the fast computation of the nonparametric maximum likelihood estimate of a mixing distribution. Journal of the Royal Statistical Society, Series B, 69, 185-198.

Liu, Y. and Wang, Y. (2018). A Fast Algorithm for Univariate Log-concave Density Estimation. *Australia & New Zealand Journal of Statistics* (To appear).

Examples

```
## Normal density
x = rnorm(1000)
(r = cnmlcd(x))

## Log-likelihood values at the NPMLE
r$ll
logLik(r$lcd, x)

## Density or log density plot
plot(r$lcd)
plot(r$lcd, x)
plot(r$lcd, x, log = TRUE)

## Density function of the Log-concave distribution
dlcd(-4:4, r$lcd)

## Estimation from log-return data
data(logreturn)
r = cnmlcd(logreturn)
plot(r$lcd, logreturn)
plot(r$lcd, logreturn, log = TRUE)

## Estimation for log-volatility data
data(logvolatility)
r = cnmlcd(logvolatility)
plot(r$lcd, logvolatility)
```

---

lcd.object *Class* lcd

Description

Class `lcd` is used to store a log-concave density function (f), where the log-density is given by a piecewise linear function.

Details

Given an `lcd` object, the density function is defined by

$$f(x;\alpha,\boldsymbol{\theta},\boldsymbol{p},L,U)=\frac{1}{C}e^{\alpha(x-L)-\sum_{j=1}^{m}p_j(x-\theta_j)_+},\quad \text{for } x\in[L,U],$$

where C is the normalizing constant given by

$$C = \int_L^U e^{\alpha(x-L) - \sum_{j=1}^m p_j (x-\theta_j)_+} \, dx$$

`coef`, `fk`, `dpk` and `cpk` can all worked out from the given parameters. They are computed when a new `lcd` object is created by function `new.lcd()` to facilitate computation when the object is used later.

Value

A list consisting of:

`alpha`	the slope of the log-density before the first interior knot.
`C`	the normalizing constant.
`theta`	vector of interior knots.
`pi`	vector of changes of slope at the interior knots.
`lower`	lower-boundary knot (L). This should be the smallest observed value.
`upper`	upper-boundary knot (U). This should be the largest observed value.
`coef`	a numeric matrix with two rows, with rows 1 and 2 storing, respectively, the intercepts and slopes of the log-density between knots.
`fk`	density values at the lower boundary (`lower`) and the interior knots (`theta`).
`dpk`	integral of x^o f(x) over each interval between knots for o = 0, 1, 2.
`cpk`	integral of x^o f(x) over each interval between the lower boundary and each knot.

See Also

`cnmlcd`, `plcd`, `dlcd`, `new.lcd`.

---

log-concave *The Log-concave Distribution.*

Description

Density and dstribution function for a log-concave distribution.

Usage

```
dlcd(x, lcd, log = FALSE)
plcd(q, lcd, lower.tail = TRUE, log.p = FALSE)
```

Arguments

x, q	vector of quantiles.
lcd	an object of class lcd.
log, log.p	logical. If TRUE, the log density or log probability values are to be returned.
lower.tail	logical. If TRUE (default), probabilities are p[X <= q]; if otherwise, p[X > q].

Value

dlcd gives the density values, and plcd gives the distribution function values.

References

Liu, Y. and Wang, Y. (2018). A Fast Algorithm for Univariate Log-concave Density Estimation. *Australia & New Zealand Journal of Statistics* (To appear).

See Also

cnmlcd, lcd.

Examples

```
x = rnorm(1000)
r = cnmlcd(x)
dlcd(-4:4, r$lcd)
dlcd(-4:4, r$lcd, log=TRUE)
plcd(-4:4, r$lcd)
plcd(-4:4, r$lcd, lower.tail = FALSE)
```

---

logreturn *Daily Log-returns of S&P 500.*

Description

252 daily log-returns of S&P 500 from 03/01/2011 to 03/01/2012.

Usage

```
logreturn
```

Format

A vector of 252 values.

Source

Data retrieved from Yahoo Finance.

See Also

cnmlcd, lcd.

Examples

```
data(logreturn)
## Computes and plots the log-concave density estimate
(r = cnmlcd(logreturn))
plot(r$lcd, logreturn)                # Density
plot(r$lcd, logreturn, log=TRUE)      # Log-density
plotgradient(r$lcd, logreturn)        # Gradient
```

---

logreturn2014 *Daily Log-returns of S&P 500 in 2014.*

Description

252 daily log-returns of S&P 500 from 02/01/2014 to 31/12/2014.

Usage

```
logreturn2014
```

Format

A vector of 252 values.

Source

Data retrieved from Yahoo Finance.

See Also

cnmlcd, lcd.

Examples

```
data(logreturn2014)
## Computes and plots the log-concave density estimate
(r = cnmlcd(logreturn2014))
plot(r$lcd, logreturn2014)              # Density
plot(r$lcd, logreturn2014, log=TRUE)    # Log-density
plotgradient(r$lcd, logreturn2014)      # Gradient
```

logvolatility *Daily Log-volatilities of S&P 500.*

Description

4783 daily log-volatilities of S&P 500 from 03/01/1995 to 03/01/2014.

Usage

```
logvolatilities
```

Format

A vector of 4786 values.

Source

Data retrieved from Yahoo Finance.

See Also

cnmlcd, lcd.

Examples

```
data(logvolatility)
## Computes and plots the log-concave density estimate
(r = cnmlcd(logvolatility))
plot(r$lcd, logvolatility)               # Density
plot(r$lcd, logvolatility, log=TRUE)     # Log-density
plotgradient(r$lcd, logvolatility)       # Gradient
```

new.lcd *Create an Object of Class "lcd"*

Description

Creates a new object of class "lcd".

Usage

```
new.lcd(alpha, theta = NULL, pi = NULL, lower, upper)
```

Arguments

alpha	the slope of the log-density before the first interior knot.
theta	vector of interior knots.
pi	vector of changes of slope.
lower	lower-boundary knot (L). This should be the smallest observed value.
upper	upper-boundary knot (U). This should be the largest observed value.

Value

An object of class lcd.

References

Liu, Y. and Wang, Y. (2018). A Fast Algorithm for Univariate Log-concave Density Estimation. *Australia & New Zealand Journal of Statistics* (To appear).

See Also

lcd, cnmlcd.

Examples

```
## Create an object of a class "lcd"
new.lcd(alpha = 1, theta = c(2,3), pi = c(1,2), lower = 0, upper = 4)
```

---

plot.lcd *Plot Log-concave Density*

Description

Plot method for class "lcd".

Usage

```
## S3 method for class 'lcd'
plot(x, data, w=NULL, log=FALSE, col="blue", knotcol=col,
   border="grey", lwd=1, pch=1, main, breaks=50, ...)
```

Arguments

x	an object of class "lcd".
data	a numeric vector storing the observations.
w	frequencies or weights for the values in x. If NULL, w = 1 is used for each value in x.
log	logical, indicating if the log density plot is to be plotted instead.

col	color of lines.
knotcol	color of knots.
border	color for the border of histogram or log histogram.
lwd	line width.
pch	point type used for knots.
main	main title.
breaks	number of break points used for creating histogram or log-histogram.
...	further arguments passed to or from other methods.

Details

Either the density or its logarithm is plotted, depending on the value of log. Knots are shown in circles. If data is provided, the plot also includes a histgram or log-histogram for data.

Author(s)

Yu Liu <liu.yu@auckland.ac.nz>, Yong Wang <yongwang@auckland.ac.nz>

References

Liu, Y. and Wang, Y. (2018). A Fast Algorithm for Univariate Log-concave Density Estimation. *Australia & New Zealand Journal of Statistics* (To appear).

See Also

cnmlcd, lcd, new.lcd, dlcd.

---

plotgradient *Plot Gradient Curve*

Description

Plots the gradient curve that corresponds to a log-concave dnesity and a data set.

Usage

```
plotgradient(lcd, x, w = NULL, col="blue", knotcol=col, pch=1,
             lwd=1, lty=1, ...)
```

Arguments

lcd	an object of class lcd.
x	a vector storing the observations.
w	a vector of frequencies (weights) of values stored in x. If w=NULL, w = 1 is used for each value in x.
col	color of lines.
knotcol	color of knots.
pch	point type for knots.
lwd	width for the gradient curve.
lty	line type for the gradient curve.
...	further arguments passed to or from other methods.

See Also

cnmlcd, lcd, new.lcd, x.weight.

Examples

```
data(logreturn)
r = cnmlcd(logreturn)
plot(r$lcd, logreturn)            # Density
plotgradient(r$lcd, logreturn)    # Gradient
```

---

x.weight *Compute Weights of Distinct Observed Values*

Description

Finds frequencies of unique observed values.

Usage

```
x.weight(x)
```

Arguments

x	observations for which frequencies (or weights) of unique values are to be found.

Value

x	vector of distinct and sorted observations.
w	vector of observation weights.

Examples

```
x.weight(c(5:1,1,3))
```

Index