Package ‘fmlogcondens’

March 23, 2018

Type Package

Title Fast Multivariate Log-Concave Density Estimation

Version 1.0.2

Description A fast solver for the maximum likelihood estimator (MLE) of a multivariate log-concave probability function. Given a sample X, it estimates a non-parametric density function whose logarithm is a concave function. Many well-known parametric densities belong to that class, among them the normal density, the uniform density, the exponential distribution and many more. This package provides functions for the estimation of a log-concave density and a mixture of log-concave densities in multiple dimensions. While being similar to the package LogConcDEAD, fmlogcondens provides much fast run times for large samples (>= 250 points). As a reference see Fabian Rathke, Christoph Schnörr (2015), <doi:10.1515/auom-2015-0053>.

License GPL (>= 2)

URL https://github.com/FabianRathke/fmlogcondens

Depends R (>= 3.2.4)

Imports geometry, MASS, mclust, mvtnorm

Encoding UTF-8

LazyData true

Suggests LogConcDEAD, knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation yes

Author Fabian Rathke [aut, cre],
Christoph Schnörr [aut],
Giovanni Garberoglio [cph]

Maintainer Fabian Rathke <frathke@gmail.com>

Repository CRAN

Date/Publication 2018-03-23 14:33:49 UTC

R topics documented:

`calcCvxHullFaces` *Calculates the convex hull of X*

Description

`calcCvxHullFaces` returns the parameters of the convex hull of X: The indices of points in X that compose the faces of conv(X) (one row per face) as well as the parameters of the hyperplanes that describe these faces.

Usage

```
calcCvxHullFaces(X)
```

Arguments

`X`	Set of data points (one sample per row)

Value

A list consisting of parameters describing the convex hull of X:

`cvh`	A matrix of indices where each row constitutes one face
`ACVH`	A matrix where each row constitutes the normal vector of a face
`bCVH`	A vector where each entry constitutes the offset of the hyperplane for a face

---

`compilationInfo` *Checks if AVX is active*

Description

`compilationInfo` is a wrapper to a C function, which prints information about whether AVX extensions where activated during compilation. Active AVX speeds up computations significantly (by a factor of 4 to 8 roughly). More informations about how AVX can be enabled during installation can be found in documentation.

Usage

```
compilationInfo()
```

`correctIntegral` *Normalizes Log-Concave Density*

Description

`correctIntegral` normalizes a log-concave density parametrized by a set of hyperplane parameters. Works by calculating y = log(f(x)) for each data point in X, normalizing y, and then recalculating a and b.

Usage

```
correctIntegral(X, mu, a, b, cvh)
```

Arguments

`X`	Set of data points (one sample per row)
`mu`	Mean vector of X that gets added back to X
`a`	Matrix where rows are hyperplane normals
`b`	Vector where entries are intercepts of hyperplanes
`cvh`	Matrix where each row is a set of indices of points in X describing one face of conv(X)

Value

Normalized hyperplane parameters (for the uncentered `X <- X + mu`)

`a, b`	Hyperplane parameters of the normalized density.
`y`	Vector with values y_i = log(f(X_)) of the normalized density.
`aSparse, bSparse`	Input hyperplane parameters.

Examples

```
# draw samples from normal distribution
X <- matrix(rnorm(200),100,2)
# calculate parameters of convex hull of X
r <- calcCvxHullFaces(X)
# draw random parameters of 10 hyperplanes
a <- matrix(runif(10*2),10,2)
b <- runif(10)

# calculate parameters of convex hull of X
params <- correctIntegral(X,rep(0,2),a,b,r$cvh)
```

---

fmlcd *Estimates a Log-Concave Density*

Description

fmlcd returns a MLE estimate of a log-concave density for X. After obtaining an initial parameter estimate the MLE objective with log-concavity and normalization constraint is optimized using a quasi-Newton approach for large scale optimization (BFGS-L). The logarithm of the optimal density f(x) is a piecewise-linear function. Its parametrization in terms of a set of hyperplanes is returned.

Usage

```
fmlcd(X, w = rep(1/nrow(X), nrow(X)), init = "", verbose = 0,
  intEps = 0.001, objEps = 1e-07, offset = 0.1, maxIter = 10000)
```

Arguments

X	Matrix of data points (one sample per row)
w	Vector of sample weights (default: uniform weights)
init	String that sets the initialization approach. 'kernel' based on kernel density, 'smooth' based on smooth log-concave density, '' compares both and takes the optimal one. (default: '')
verbose	Int determining the verboseness of the code; 0 = no output to 3. (default: 0)
intEps	Stopping criteria for the numerical integration accuracy Optimization stops if both measurements are smaller than intEps and objEps Modification of this value is not recommended. (default: 1e-3)
objEps	Stopping criteria for the change in the objective function Optimization stops if both measurements are smaller than intEps and objEps Modification of this value is not recommended. (default: 1e-7)
offset	Smaller values correspond to slower hyperplane reduction. Offset should be a value smaller than 1. Modification of this value is not recommended. (default: 1e-1)
maxIter	Number of iterations in the main optimization (default: 1e4)

Value

Parametrization of f(x) in terms of hyperplanes and function evaluations y = log(f(x))

`aOpt, bOpt` Analytically normalized parameters of f(x).

`logLike` Log-likelihood of f(x)

`y` Vector with values y_i = log(f(X_)) of the normalized density ($logLike = \sum(y_i)$).

`aOptSparse, bOptSparse` Sparse parametrization normalized on the integration grid.

Examples

```
# draw samples from normal distribution
X <- matrix(rnorm(200),100,2)
# calculate parameters of convex hull of X
r <- calcCvxHullFaces(X)
# draw random parameters of 10 hyperplanes
a <- matrix(runif(10*2),10,2)
b <- runif(10)

# calculate parameters of convex hull of X
params <- correctIntegral(X,rep(0,2),a,b,r$cvh)
```

---

fmlcdEM *Estimates a Log-Concave Mixture Density*

Description

`fmlcdEM` Utilizes the EM approach to obtain a mixture of log-concave densities. Utilizes Gaussian hierarchical clustering to initilize the posterior probabilities of class affiliation (as proposed by the package LogConcDEAD by Cule et al.).

Usage

```
fmlcdEM(X, K = 2, posterior, verbose = 0, maxIter = 50)
```

Arguments

`X` Matrix of data points (one sample per row)

`K` Number of latent variables (default: 2)

`posterior` Matrix with posterior probabilities for class affiliation; Initialized if not provided using a Gaussian hierarchical clustering.

`verbose` Int determining the verboseness of the code; 0 = no output to 3. (default: 0)

`maxIter` Maximal number of EM iterations. (default: 50)

Value

Parametrization of the mixture density

`params`	List of length K, where each entry contains the hyperplane for one density
`densEst`	Matrix where each row contains the marginal distribution p(x)
`tau`	Marginal distribution over the latent variable p(z)

`paramFitGammaOne` *Parameter Initialization Based on a Smooth Log-Concave Density*

Description

`paramFitGammaOne` fits in a first step a log-concave density to data points X with weights w using a smoothness parameter of gamma=1. In a second step, it calculates the upper convex hull of the set X and log(y), where y_i is the smooth log-concave density evaluated at X_i. It returns the hyperplane parameters of the faces of this upper convex hull.

Usage

```
paramFitGammaOne(X, w, ACVH, bCVH, cvh)
```

Arguments

`X`	Set of data points (one sample per row)
`w`	Vector with weights for X (`sum(w)==1`)
`ACVH`	Matrix where each row constitutes the normal vector of a face of conv(X)
`bCVH`	Vector where each entry constitutes the intercept for a face of conv(X)
`cvh`	Matrix where each row is a set of indices of points in X describing one face of conv(X)

Value

A list containing the description of the upper convex hull of (X,log(y)) in term of hyperplane parameters:

`a`	A matrix where each row constitutes a hyperplane normal
`b`	A vector where each entry constitutes the intercept of a hyperplane

Note the difference

Examples

```
# draw samples from normal distribution
X <- matrix(rnorm(200),100,2)
# calculate parameters of convex hull of X
r <- calcCvxHullFaces(X)
# find initial hyperplane parameters based on a smooth log-concave density
params <- paramFitGammaOne(X, rep(1 / nrow(X), nrow(X)), r$ACVH, r$bCVH, r$cvh)
```

paramFitKernelDensity *Parameter Initialization Based on a Kernel Density*

Description

`paramFitKernelDensity` first fits a kernel density to a sample X with weight vector w. It then calculates the parameters of the piecewise linear function defined to be the upper convex hull of (X,log(y)).

Usage

```
paramFitKernelDensity(X, w, cvh, h = apply(X, 2, sd) * n^(-1/(d + 4)))
```

Arguments

`X`	Set of data points (one sample per row)
`w`	Vector with weights for X (`sum(w) == 1`)
`cvh`	Matrix where each row is a set of indices of points in X describing one face of conv(X)
`h`	Scalar parameter that governs the Gaussian kernel

Value

A list containing the description of the upper convex hull of (X,log(y)) in term of hyperplane parameters:

`a`	A matrix where each row constitutes the normal vector of a face
`b`	A vector where each entry constitutes the offset of a face

Examples

```
# draw samples from normal distribution
X <- matrix(rnorm(200),100,2)
# calculate parameters of convex hull of X
r <- calcCvxHullFaces(X)
# find initial hyperplane parameters based on a kernel density estimator with Gaussian kernel
params <- paramFitKernelDensity(X, rep(1 / nrow(X), nrow(X)), r$cvh)
```

Index