Package ‘UEM’

July 1, 2020

Type Package

Title Estimating and updating mixtures via EM

Version 0.3-1

Date 2020-06-30

Author Jochen Einbeck, Daniel Bonetti, and Najla Qarmalah

Maintainer Jochen Einbeck <jochen.einbeck@durham.ac.uk>

Depends R (>= 3.5.0)
Imports mvtnorm (>= 1.0-9)
Suggests qcc, npmlreg


License GPL (>=2)

NeedsCompilation no

R topics documented:

  draw1d.umix ..................... 1
  EM ................................ 2
  energy1 .......................... 4
  estep ............................ 5
  gaussSim .......................... 5
  init ............................. 6
  kboxplot .......................... 7
  logLike ........................... 9
  plot.umix .......................... 10
  UEM ................................ 11
  wtSigma ........................... 12

<table>
<thead>
<tr>
<th>draw1d.umix</th>
<th>Auxiliary drawing functions</th>
</tr>
</thead>
</table>

Description

Functions to plot fitted mixtures
EM

Fitting and updating mixtures via the EM algorithm

Description

The function EM implements the usual EM algorithm for multivariate Gaussian mixtures or Poisson mixtures. The function updateEM allows for updating the result from a fitted mixture model, after new observations have been coming in. The function UEM allows to split a data set (from the start) into several batches, and apply and update EM sequentially on those batches.

Usage

EM(y, K, init = "quantile", family = "Gaussian", iter = -1, threshold = 0.0001, lambda = 0.999, tol=0.5, verbose = FALSE, plot = FALSE, ...)  
UEM(y, K, init = "quantile", split = NULL, randomize = FALSE, family = "Gaussian", iter = -1, threshold = 0.001, max.time = NULL, lambda = 0.999, verbose = FALSE, plot = FALSE, ...)  
updateEM(z = NULL, theta, iter = -1, threshold = 0.001, max.time = NULL, lambda = 0.999, verbose = FALSE, plot = FALSE, ...)

Arguments

y  a univariate or multivariate data set.
z  new data to be added for updating the estimate theta.
theta  the value of the parameter vector theta from which UEM is started.
K  the number of components.
init  the type of initialization used. Options include "random" (randomly chosen from the data), "scatter" (uniformly sampled from the support of the data), "quantile" (quantile-based), "shortruns" (short runs of EM), "gq" (based on Gauss Quadrature points). See also help file for init.
EM

split
For the use in UEM, a vector giving the split points between the batches.

randomize
Boolean. For the use in UEM. If TRUE, data are randomized before splitting.

family
Response family. At present, "Gaussian" (default) and "Poisson" are supported. "Cauchy" is in preparation.

iter
Number of EM iterations. For UEM this can be a vector which gives the number of iterations in the individual batches (in this case its length has just to be the length of split+1) or a scalar (in this case all batches have the same size).

threshold
Convergence threshold (in terms of a log-likelihood difference).

lambda
calibrates between globally equal component variances (\(\lambda=0\)) or unequal variances (\(\lambda=1\)). The only reason to set this to a value different than 0 or 1 is to avoid likelihood spikes when using unequal variances. In this case a value like 0.999 is suitable, which means that the component variances are computed by taking 0.999 times the component-specific (‘unequal’) variances plus 0.001 the globally computed (‘equal’) component variance.

tol
tuning parameter which scales EM starting points inwards or outwards

max.time
Time limit after which execution stops (in seconds).

verbose
Boolean. If TRUE, displays iteration count on the screen.

plot
Boolean. If TRUE, provides graphical output (fitted mixture).

... Arguments to be passed to plot.

Value
A fitted mixture object, of class umix.

Author(s)
J. Einbeck, D. Bonetti

References

Examples
### Univariate Gaussian Example:

```r
data(pistonrings, package="qcc")
boxplot(diameter ~ sample, data=pistonrings)

dm <- as.matrix(pistonrings$diameter)

# EM all at once:
fit <- EM(dm,2, threshold=0.005)

# Now via update EM
fit2 <- UEM(dm,2, split=seq(100,200,by=5), iter= c(10, rep(2,20),-1), plot=TRUE )

# (this gives 100 data points first and iterates 10 times. Then it gives 20 batches
# of size five and iterates twice after each batch. Finally it iterates until convergence).
```
# Compare log-likelihoods:
logLike(fit, dm)
logLike(fit2, dm)

### Bivariate Gaussian Example:

```r
require(mvtnorm)
s1 <- matrix(c(10,3,3,2),2,2)
s2 <- matrix(c(1,3,3,16),2,2)
m1 = rnorm(n=40, c(4,2), s1)
m2 = rnorm(n=60, c(9,4), s2)
x = rbind(m1, m2)
par(mfrow=c(2,2))
plot(x)

thetar = EM(x, 2, iter=10) # Standard EM
plot(thetar, x, main="EM")

i = sample(100, 50)
theta0 = EM(x[-i, ], 2, iter=10) # remove 50 points, fit EM to remaining points
theta1 = updateEM(x[i,], theta0, iter=10) # put points back, update EM

plot(theta0, x[-i, ], col=1, main= "EM (subset)"
plot(theta1, x, col= 1 + (1:100)%in%i, main = "update EM")
```

### Poisson Example:

```r
theta <- list("mu"=c(1,8,30),"pi"=c(0.2,0.5,0.3))
theta2 <- list("mu"=c(5,10,100),"pi"=c(0.2,0.2,0.6))
pdat <- poisSimN(100, theta)
pdat.z <- poisSimN(20, theta2)
poisfit <- EM(pdat, 3, iter=100, family="Poisson")
plot.umix(poisfit, pdat)
poisup <- updateEM(pdat.z, poisfit, iter=100, dist="Poisson", plot=TRUE)
# equivalently, at once:
poisall <- UEM(c(pdat, pdat.z), 3, split=100, iter=100, family="Poisson", plot=TRUE)
poisup$mu
poisall$mu
# identical!
```

data("energy1")
data("energy2")
Format

`energy1` is a data frame giving the energy use (kg of oil equivalent per capita) for 134 countries for the years 1971 to 2011. `energy2` is a log10 version of the original data frame.

Details

Energy use refers to use of primary energy before transformation to other end-use fuels, which is equal to indigenous production plus imports and stock changes, minus exports and fuels supplied to ships and aircraft engaged in international transport.

Source


References


Examples

```r
data(energy2)
boxplot(energy2[,c(1,6,11,16,21,26,31,36,41)],xlab="year",ylab="log energy use")
```

estep                  E-step and M-step

Description

Expectation and Maximization steps.

Usage

```r
estep(theta, y)
mstep(y, W, lambda, family = "Gaussian")
```

Arguments

- `theta`
- `y`
- `W`
- `lambda`
- `family`

Author(s)

J. Einbeck, D. Bonetti
gaussSim  

_Simulating from mixture models_

**Description**

Functions to simulate data from Gaussian, Poisson, and Cauchy mixtures.

**Usage**

```r
gaussSim(theta)
gaussSimN(n, theta)
poisSim(theta)
poisSimN(n, theta)
cauchSim(theta)
cauchSimN(n, theta)
```

**Arguments**

- `theta` a list with elements `pi` (vector of mixture proportions), `mu` (a matrix where each row is one mean vector), and `Sigma` (a list of as many variance matrices as mixture components). In the case of the Cauchy model, `Sigma` is a list of gamma parameters.
- `n` sample size

**Author(s)**

J. Einbeck, D. Bonetti, Z. Kalantan

**Examples**

```r
rho <- 0.7
Sigma <- list(diag(c(1,2)), matrix(c(2, 2*rho, 2*rho, 1), byrow=TRUE, ncol=2))
theta <- list(pi=c(1/3, 2/3), mu=matrix(c(3,1,0,4), byrow=TRUE, ncol=2), Sigma=Sigma)
sim <- gaussSimN(1000, theta)

fit2 <- EM(sim, K=2)
plot.umix(fit2)

sim1 <- cauchSimN(100, theta=list(pi=rep(0.5,2), mu=c(0,20), Sigma=list(gamma=1, gamma=2)))
hist(sim1, breaks=40)
```

**init**  

_Initializing the EM algorithm_

**Description**

Functions to initialize the EM algorithm.
Usage

init(y, K, type = "quantile", tol=0.5, ...)  
shortruns(y, K, init = "random", family = "Gaussian", maxit = 50,  
          threshold = 10, lambda = 1, verbose = TRUE, plot = FALSE)

Arguments

y  
a univariate or multivariate data set.
K  
the number of components.
type  
type of starting points. Includes "random" (randomly chosen from the  
data), "scatter" (uniformly sampled from the support of the data), "quantile" (quantile-based), "shortruns" (short runs of EM), "gq" (based on Gauss Quadrature points).
tol  
only relevant for type="gq". Scales Gaussian Quadrature points inwards or outwards by the factor tol.
init
family
maxit
threshold
lambda
verbose
plot
...

Author(s)

J. Einbeck and D. Bonetti

kboxplot

\textit{k-boxplots for mixture data}

Description

k-boxplots visualize the k components of mixture models by k different boxes (Qarmalah, Einbeck & Coolen, 2016).

Usage

kboxplot(data, W=NULL, k, type="default", cen=0, colbox,  
         xlim, ylim, col, xlab, ylab, xaxt, main=type )
## Arguments

- **data**
  - A univariate data set for which a k-boxplot is to be produced. NAs are allowed in the data.

- **W**
  - The responsibility matrix or weight matrix. This is a n x k matrix, where n is the length of the data set and k the number of mixture components. If the matrix W is not provided, it will be computed using EM.

- **k**
  - The number of components.

- **type**
  - Specifies the way in which observations outside the boxes are displayed. Possible types are "plain", which simply draws whiskers until the maximum and minimum observations, "default" which displays individual points coloured by MAP classification, "full" for drawing lines which display the posterior possibilities. In addition, for k=2 only the option "two" is supported for drawing colored lines on both sides of the boxes. The boxes are drawn in exactly the same way under all four options.

- **cen**
  - A real number on the x-axis at which the k-boxplots are centered.

- **colbox**
  - Color(s) to fill or shade the rectangle(s) with. The default NA (or also NULL) means do not fill.

- **xlim, ylim**
  - Numeric vectors of length 2, giving the x and y coordinates ranges. By default, they are (xlim = cen + c(-1,1)) and (ylim = c(min(data), max(data)) + c(-1,1)*0.1) respectively.

- **col**
  - If col is not missing it is assumed to contain colors to be used to colour the borders of the boxes, lines and points. By default they are colored using the command rainbow(k).

- **xlab**
  - A title for the x axis: see title.

- **ylab**
  - A title for the y axis: see title.

- **xaxt**
  - A character which specifies the x axis type. Specifying "n" suppresses plotting of the axis. The standard value is "s": for compatibility with S values "l" and "t" are accepted but are equivalent to "s": any value other than "n" implies plotting.

- **main**
  - An overall title for the plot

## Details

The k-boxplot is a new plot tailored to mixture data, where k is the number of mixture components. It visualizes the k components of mixture models by k different boxes, compared to a boxplot which has only one box. Then, a boxplot is a special case of a k-boxplot when k=1. Bottom and top of the boxes are drawn at the weighted first and third quartiles of the data in each group respectively. Weighted medians are displayed as horizontal lines drawn inside the boxes. Furthermore, optionally, the posterior probabilities of group membership can be visualised by appropriate lines and points. The required information in order to draw a k-boxplot can be estimated by different methods, for example by the EM-algorithm.

## Value

A plotted k-boxplot

## Author(s)

N. Qarmalah and J. Einbeck
References


See Also

EM

Examples

# This code can be used to reproduce all examples in Qarmalah, Einbeck and Coolen (2016).

# Energy use data:
data(energy2)
eng<-energy2[,"2011"]
W<-EM(eng,2)$W
par(mfrow=c(2,2))
kboxplot(eng,W,2, xlab="2011", ylab="log energy use", type="plain")
kboxplot(eng,W,2, xlab="2011", ylab="log energy use", type="default")
kboxplot(eng,W,2, xlab="2011", ylab="log energy use", type="full")
kboxplot(eng,W,2, xlab="2011", ylab="log energy use", type="two")

# Internet users data
data(WWWusage)
par(mfrow=c(1,2))
E3 <- EM(log(WWWusage),3, lambda=1, init="gq", tol=2) # unequal component variances
kboxplot(log(WWWusage),E3$W,3,main="(a)", type="default")
E3a<- EM(log(WWWusage),3, lambda=0) # equal component variances
kboxplot(log(WWWusage),E3a$W,3,main="(b)", type="default")

E4<- EM(log(WWWusage),4, lambda=1, init="gq", tol=2) # unequal component variances
kboxplot(log(WWWusage),E4$W,4,main="(a)", type="full")
E4a<- EM(log(WWWusage),4, lambda=0) # equal component variances
kboxplot(log(WWWusage),E4a$W,4,main="(b)", type="full")

# Toxoplasmosis (rainfall) data
require(npmlreg)
data(rainfall)
toxo.np3<- alldist(cbind(Cases,Total-Cases) ~ 1, random=~1, random.distribution="np", family=binomial(link=logit), data=rainfall,k=3, plot.opt=0, verbose=FALSE)
W <- post(toxo.np3)$prob
par(mfrow=c(1,2))
kboxplot(rainfall$Cases/rainfall$Total, W, ylim=c(0,0.75), main="cases/total")
kboxplot(toxo.np3$fitted, W, ylim=c(0,0.75), main="fitted")
**logLike**

*Log-likelihood of fitted model*

**Description**

Produces numerical value of the log-likelihood of a fitted model.

**Usage**

```r
logLike(theta, y)
```

**Arguments**

- `theta`
- `y`

**Examples**

```r
## Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##-- or do help(data=index) for the standard data sets.
## The function is currently defined as
function (theta, y, K)
{
  if (is.vector(y)) {
    y <- matrix(y, ncol = 1)
  }
  if (is.null(theta$family)) {
    if (!is.null(theta$var)) {
      theta$family <- "Gaussian"
    } else {
      theta$family <- "Poisson"
    }
  }
  dens <- matrix(0, dim(y)[1], K)
  for (k in 1:K) {
    dens[, k] <- switch(theta$family, Gaussian = theta$pi[k] *
                        dmvnorm(y, mean = theta$mu[k, ], sigma = theta$var[k]),
                        Poisson = theta$pi[k] * dpois(y, lambda = theta$mu[k])
                      )
  }
  loglik <- sum(log(apply(dens, 1, sum)))
  return(loglik)
}
```
plot.umix  

Plotting function for umix objects

Description

Generic plotting function for umix objects.

Usage

## S3 method for class 'umix'
plot(x, y = NULL, showMeans = TRUE, contours = TRUE, col = "cornflowerblue", ...)

Arguments

x  
an umix object, often denoted as theta.
y  
original data; can also be given as component of x.
showMeans
contours
col
...

Author(s)

J. Einbeck

Examples

data(rock)
fit <- EM(rock[,2:3], K=2)
plot(fit)

UEM  

Estimating and updating mixtures via EM

Description


Details

The DESCRIPTION file:

Package: UEM
Type: Package
Title: Estimating and updating mixtures via EM
Version: 0.3-1
Date: 2020-06-30
Author: Jochen Einbeck, Daniel Bonetti, and Najla Qarmalah
Maintainer: Jochen Einbeck ¡jochen.einbeck@durham.ac.uk¿
wtSigma

Index of help topics:

<table>
<thead>
<tr>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimating and updating mixtures via EM</td>
</tr>
<tr>
<td>Fitting and updating mixtures via the EM algorithm</td>
</tr>
<tr>
<td>draw1d.umix</td>
</tr>
<tr>
<td>Auxiliary drawing functions</td>
</tr>
<tr>
<td>energy1</td>
</tr>
<tr>
<td>Energy use data</td>
</tr>
<tr>
<td>estep</td>
</tr>
<tr>
<td>E-step and M-step</td>
</tr>
<tr>
<td>gaussSim</td>
</tr>
<tr>
<td>Simulating from mixture models</td>
</tr>
<tr>
<td>init</td>
</tr>
<tr>
<td>Initializing the EM algorithm</td>
</tr>
<tr>
<td>kboxplot</td>
</tr>
<tr>
<td>k-boxplots for mixture data</td>
</tr>
<tr>
<td>logLike</td>
</tr>
<tr>
<td>Log-likelihood of fitted model</td>
</tr>
<tr>
<td>plot.umix</td>
</tr>
<tr>
<td>Plotting function for 'umix' objects</td>
</tr>
<tr>
<td>wtSigma</td>
</tr>
<tr>
<td>Auxiliary functions</td>
</tr>
</tbody>
</table>

The package bundles two pieces of software resulting from publications as listed below;

1. Functions EM, UEM, and updateEM to fit Gaussian and Poisson mixtures as a whole, or sequentially as an update algorithm
2. A function k-boxplot to produce and display k-boxplots for mixture data

Author(s)

Jochen Einbeck, Daniel Bonetti, and Najla Qarmalah

Maintainer: Jochen Einbeck ¿jochen.einbeck@durham.ac.uk¿

References


Description

Auxiliary functions (not to be called by user)
wtSigma

Usage

wtSigma(s, wk, K)
dkern2(K, lambda)
Max(x)
weighted.quantile.top(dat, weights, p)
weighted.quantile(dat, weights, p)
WQ(x,y)
Lines(dat,z,WIK,cen,coll,type)

Arguments

s
wk
K
lambda
x
dat
weights
p
y
z
WIK
cen
coll
type