

Package ‘LCTMtools’

August 26, 2019

Type Package

Title Latent Class Trajectory Models: Tools for checking adequacy

Version 0.1.3

URL <http://www.hannahlennon.com>

BugReports <https://github.com/hlennon/LCTMtools/issues>

Description A selection of model adequacy tests for Latent Class Trajectory Models (LCTMs) which include the APPA (average posterior probability of assignment), the OCC (odds of correct classification), entropy and relative entropy.
See the github.com/hlennon/LCTMtools for more information, documentation and examples.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Depends R (>= 2.10)

Imports stats,
psych,
lcmm,
ggplot2,
dplyr

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

actual_proportions	2
appa	2
bmi	3
bmi_long	3
class_assignment	4
confusion_matrix	4
entropy	5
gg_color_hue	6
kappa_matrix	6
LCTMcompare	7

LCTMtoolkit	8
LCTMtools	9
mismatch	9
mmlcr_to_lctm	10
occ	10
plotLCTM	11
relative_entropy	11
residualplot_step1	12
sastraj_to_lctm	13
Index	14

actual_proportions	<i>The proportion within each class</i>
--------------------	---

Description

actual_proportions The proportion within each class AFTER class assignment (using max posterior rule)

Usage

actual_proportions(p)

Arguments

p is the posterior probabilities of assignment of dimensions, K columns and N rows

Value

The proportion within each class AFTER class assignment (using max posterior rule)

Examples

Not run: actual_proportions(p)

appa	<i>The Average Posterior Probability Assignment (APPA)</i>
------	--

Description

appa

Usage

appa(p)

Arguments

p is the posterior probabilities of assignment of dimensions, K columns and N rows

Details

Computes the Average Posterior Probability Assignment (APPA) for a K latent class trajectory model.

Value

The Average Posterior Probability Assignment (APPA) for each class

Examples

```
## Not run: appa(p)
```

bmi	<i>Body Mass Index (BMI) repeated measures of 10,000 individuals in wide format.</i>
-----	--

Description

A wide format data frame of simulated BMI values of 10,000 individuals.

Usage

```
bmi
```

Format

A wide format data frame of simulated BMI values of 10,000 individuals.

id Individual ID

bmi Body mass index of the individual at times T1,T2, T3 and T4, in kg/m²

T Time of BMI measure, in years

true_class Tag to identify the class the individual BMI data was simulated from

bmi_long	<i>Body Mass Index (BMI) repeated measures of 10,000 individuals in long format.</i>
----------	--

Description

A long format data frame of simulated BMI values of 10,000 individuals.

Usage

```
bmi_long
```

Format

A long format data frame of simulated BMI values of 10,000 individuals.

id Individual ID

age Age of BMI measure, in years

bmi Body mass index of the individual at times T1,T2, T3 and T4, in kg/m²

true_class Tag to identify the class the individual BMI data was simulated from

class_assignment	<i>Maximum class assignment class_assignment Maximum class assignment</i>
------------------	---

Description

Maximum class assignment class_assignment Maximum class assignment

Usage

```
class_assignment(p)
```

Arguments

p is the posterior probabilities of assignment of dimensions, K columns and N rows

Value

This function computes the Average Posterior Probability Assignment (APPA) for a K latent class trajectory model.

Examples

```
## Not run: class_assignment(p)
```

confusion_matrix	<i>A confusion matrix</i>
------------------	---------------------------

Description

A matrix, also known as a matching matrix or an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one. Each row of the matrix represents the instances in a predicted class for model 1 while each column represents the instances in class for model 2. The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabeling one as another).

Usage

```
confusion_matrix(model1, model2, "ModelA", "ModelB")
```

Arguments

model1	A fitted model from the lcmm R package (or from SAS passed through the SASmodelbuilder() function)
model2	is the posterior probabilities of assignment of dimensions, K columns and N rows
name1	optional paramter to pre-specify name of model
name2	optional paramter to pre-specify name of model

Value

A confusion matrix between two models with the same number of classes

Examples

```
## Not run:
data(bmi_long, package='LCTMtools')
library(lcmm)
model1 <- lcmm::hlme(BMI ~Age,
  mixture= ~Age,
  random= ~Age,
  nwg=TRUE, ng=2, subject='ID', data=data.frame(bmi_long[1:500, ]))
model2 <- lcmm::hlme(BMI ~Age,
  mixture= ~Age,
  random= ~1,
  nwg=FALSE, ng=2, subject='ID', data=data.frame(bmi_long[1:500, ]))
confusion_matrix(model1, model2)
## End(Not run)
```

entropy

Entropy

Description

A global measure of uncertainty with values close to zero implying a good model. Entropy is a global measure of classification uncertainty, which takes into account all $N \times K$ posterior probabilities. The entropy of a model is defined as which takes values from $[0, \text{infinity})$, with higher values indicating a larger amount of uncertainty. Entropy values closest to 0 correspond to models with least classification uncertainty.

Usage

```
entropy(p)
```

Arguments

p	is the posterior probabilities of assignment of dimensions, K columns and N rows
---	--

Value

Entropy value between (0, infinity)

Examples

```
## Not run: entropy(p)
```

gg_color_hue	<i>gg_color_hue</i>
--------------	---------------------

Description

Emulate ggplot2 default colour palette

Usage

```
gg_color_hue(n)
```

Arguments

n	Number of colours
---	-------------------

Value

a vector of n colour codes

References

https://www.rdocumentation.org/packages/iprior/versions/0.7.1/topics/gg_colour_hue

Examples

```
gg_color_hue(2)
gg_color_hue(4)
plot(1:10, pch=12, col=gg_color_hue(10), lwd=30, xaxt="n", yaxt="n", ylab="", xlab="")
```

kappa_matrix	<i>Kappa matrix</i>
--------------	---------------------

Description

Kappa matrix of cohen's kappa values

Usage

```
kappa_matrix(ConfMatrix, acc = 2)
```

Arguments

ConfMatrix	A confusion matrix made using confusion_matrix(model1, model2)
acc	The accuracy of the results, defaults to 2 decimal places

Value

Unweighted and weighted Kappa value computed using the `cohen.kappa()` function from the `psych` R package

Examples

```
data(bmi_long, package='LCTMtools')
require(lcmm)
require(psych)
model1 <- hlme(fixed = bmi ~ age,
mixture= ~ age,
random= ~ age,
nwg=TRUE, ng=2, subject="id", data=data.frame(bmi_long[1:500, ]))

model2 <- hlme(fixed = ~ age,
mixture= ~ age,
random= ~1, nwg=FALSE, ng=2, subject="id", data=data.frame(bmi_long[1:500, ]))
ConfMatrix <- confusion_matrix(model1, model2)
kappa_matrix(ConfMatrix)
```

LCTMcompare

A model comparison toolkit. LCTMcompare

Description

The function `LCTMcompare` gives a summary of comparison between fitted LCTM models.

Usage

```
LCTMcompare(modelA, modelB)
```

Arguments

<code>modelA</code>	is the output from <code>hlme()</code> R model or <code>model</code> is the output of <code>SASmodelbuilder(oe, os, op, of)</code> passed through it
<code>modelB</code>	the model to be compared which is the output from <code>hlme()</code> R model or <code>model</code> is the output of <code>SASmodelbuilder(oe, os, op, of)</code> passed through it

Value

A selection of model adequacy tests, including the APPA (average posterior probability of assignment), the OCC (odds of correct classification), entropy SE , Relative entropy (SE_k),

References

<https://bmjopen.bmj.com/content/8/7/e020683>

Examples

```
data(bmi_long, package='LCTMtools')
require(lcmm)
set.seed(999)
data(bmi_long, package = 'LCTMtools' )
# Use the hlme function from the 'lcmm' R package to fit a 2 class latent class trajectory model
model2classes <- lcmm::hlme(fixed = bmi ~ age + I(age^2),
                           mixture= ~ age,
                           random = ~ age,
                           ng = 2,
                           nwg = TRUE,
                           subject = "id",
                           data = data.frame(bmi_long[1:500, ] ))
# Compute model adequacy measures
LCTMtoolkit(model2classes)
# Compare with a 3 class model
model3classes <- lcmm::hlme(fixed = bmi ~ age + I(age^2),
                           mixture= ~ age,
                           random = ~ age,
                           ng = 3,
                           nwg = TRUE,
                           subject = "id",
                           data = data.frame(bmi_long[1:500, ] ))
LCTMtoolkit(model3classes)
LCTMcompare(model2classes, model3classes)
```

LCTMtoolkit

A toolkit which computes a selection of model adequacy tests
LCTMtoolkit

Description

The function LCTMtoolkit computes a selection of model adequacy tests, including the APPA (average posterior probability of assignment), the OCC (odds of correct classification), entropy E, Relative entropy (E_k), odds of correct classification is the ratio of the odds of classification based on the maximum posterior probability classification rule and the estimated class membership proportions (π_k)

Usage

```
LCTMtoolkit(model)
```

Arguments

model	the models to be compared which is the output from hlme() R model or model is the output of SASmodelbuilder(oe, os, op, of) passed through it
-------	---

Value

A selection of model adequacy tests, including the APPA (average posterior probability of assignment), the OCC (odds of correct classification), entropy EE , Relative entropy (EE_k),

References

<https://bmjopen.bmj.com/content/8/7/e020683>

Examples

```
data(bmi_long, package='LCTMtools')
require(lcmm)
model2class <- lcmm::hlme(fixed = bmi ~ age,
mixture= ~ age,
random= ~ age,
nwg=TRUE, ng=2, subject="id",
data=data.frame(bmi_long[1:500, ]))
postprob(model2class)
LCTMtoolkit(model2class)
```

LCTMtools	<i>LCTMtools: A package for computing a number of Latent Class Trajectory Model tools for a given hlme() object or SAS model.</i>
-----------	---

Description

The LCTMtools package provides two categories of important functions: LCTMtools (to test a models adequacy) and LCTMcompare (to aid model selection).

LCTMtools functions

The LCTMtools functions arw a selection of model adequacy tests for Latent Class Trajectory Models (LCTMs) which include the APPA (average posterior probability of assignment), the OCC (odds of correct classification), entropy, Relative entropy.

mismatch	<i>Computes the mismatch of the posterior probabilities (mismatch=actual-estimated)</i>
----------	---

Description

Computes the mismatch of the posterior probabilities (mismatch=actual-estimated)

Usage

```
mismatch(p, pi)
```

Arguments

p	is the posterior probabilities of assignment of dimensions, K columns and N rows
pi	is the estimated proportion of class membership of length K

Value

The mismatch of posterior probabilities

Examples

```
## Not run: mismatch(p, pi)
```

mmlcr_to_lctm	<i>Converts an R mmlcr model output to the format of R's hlme class</i>
---------------	---

Description

Converts an R mmlcr model output to the format of R's hlme class

Usage

```
mmlcr_to_lctm(model)
```

Arguments

model contains model parameter estimates and maximised likelihood, AIC, BIC values

Value

A format to feed into the LCTMtoolkit() R function

Examples

```
## Not run: mmlcr_to_lctm(model)
```

occ	<i>The odds of correct classification is the ratio of the odds of classification based on the maximum posterior probability classification rule and the estimated class membership proportions (pi_k).</i>
-----	--

Description

The odds of correct classification is the ratio of the odds of classification based on the maximum posterior probability classification rule and the estimated class membership proportions (pi_k).

Usage

```
occ(p, pi)
```

Arguments

p is the posterior probabilities of assignment of dimensions, K columns and N rows

pi is the estimated proportion of class membership of length K

Value

The odds of correct classification

Examples

```
## Not run: occ(p, pi)
```

plotLCTM

*plotLCTM***Description**

A wrapper function to plot hlme trajectories in ggplot2 style. This does the same function as predictY in the lcmm package.

Usage

```
plotLCTM(m, shape, xlimit = c(0, 4.7), ylimit = c(20, 40),
  splinesnewdata = NULL)
```

Arguments

m fitted hlme or lcmm model using the lcmm R package

Value

A plot in ggplot style

Examples

```
library(ggplot2)
data(bmi_long, package='LCTMtools')
require(lcmm)
model2class <- lcmm::hlme(fixed = bmi ~ age,
  mixture= ~ age,
  random= ~ age,
  nwg=TRUE, ng=2, subject="id",
  data=data.frame(bmi_long[1:500, ]))
plotLCTM(model2class, shape="linear")

library(splines) # For use of natural splines
model2class_splines <- lcmm::hlme(fixed = bmi ~ ns(age, knots=2),
  mixture= ~ ns(age, knots=2),
  random= ~ age,
  nwg=TRUE, ng=2, subject="id",
  data=data.frame(bmi_long[1:500, ]))
newdat <- data.frame(age=seq(0, 4.7, length=100))
#plotLCTM(model2class_splines, shape="splines", splinesnewdata = newdat,
#xlimit=c(0, 4.7), ylimit=c(20, 40))
```

relative_entropy

*The Relative Entropy***Description**

The Relative Entropy

Usage

```
relative_entropy(p)
```

Arguments

`p` is the posterior probabilities of assignment of dimensions, K columns and N rows

Value

Relative Entropy - where values close to 1 indicate lowest classification uncertainty. In the special case when there is most uncertainty and each individual has equal probability of belonging to each class, $E_K=0$. Jedidi et al., describes relative entropy as a relative measure of ‘fuzziness’, and suggested cause concern when close to zero, as this implies that the latent class centroids are not sufficiently separated.

Examples

```
## Not run: relative_entropy(p)
```

```
residualplot_step1      residualplot_step1
```

Description

A wrapper function to implement step 1 of the 8 step framework. This is a wrapper function to the lcmm/hlme model fit to examine the class-specific residuals in order to aid choice of random effect distribution.

Usage

```
residualplot_step1(model, nameofoutcome = "bmi", nameofage = "age",
  data = bmi_long, ylimit = c(-5, 5))
```

Arguments

`model` fitted hlme or lcmm model using the lcmm R package
`nameofoutcome` Name of the longitudinal variable in the dataset
`nameofage` Name of the age variable in the dataset
`data` Name of the dataframe (long format) used for the regression fit
`type` Type of residual plots: lines (spaghetti) or points

Value

Class-specific residual plots in ggplot style

Examples

```
library(ggplot2)
data(bmi_long, package = "LCTMtools")
require(lcmm)
model2class <- lcmm::hlme(
  fixed = bmi ~ age,
  mixture = ~age,
  random = ~ -1,
  nwg = TRUE, ng = 2, subject = "id",
  data = data.frame(bmi_long[1:500, ])
)
residualplot_step1(model2class,
  nameofoutcome = "bmi",
  nameofage = "age",
  data = bmi_long,
)
```

sastraj_to_lctm

*Converts a SAS proc traj model to the format of R's hlme class***Description**

Converts a SAS proc traj model to the format of R's hlme class

Usage

```
sastraj_to_lctm(oe, of, op, os)
```

Arguments

oe	contains model parameter estimates and maximised likelihood, AIC, BIC values
of	contains posterior probabilities
op	contains predictors
os	contains fixed effect and class membership parameter estimates

Value

A format to feed into the LCTMtoolkit() R function

Examples

```
## Not run: sastraj_to_lctm(oe, of, op, os)
os$PI/100
exp(c(M$best[1:4], 0))/(sum(exp(c(M$best[1:4], 0))))

## End(Not run)
```

Index

*Topic **datasets**

bmi, [3](#)

bmi_long, [3](#)

actual_proportions, [2](#)

appa, [2](#)

bmi, [3](#)

bmi_long, [3](#)

class_assignment, [4](#)

confusion_matrix, [4](#)

entropy, [5](#)

gg_color_hue, [6](#)

kappa_matrix, [6](#)

LCTMcompare, [7](#)

LCTMtoolkit, [8](#)

LCTMtools, [9](#)

LCTMtools-package (LCTMtools), [9](#)

mismatch, [9](#)

mmlcr_to_lctm, [10](#)

occ, [10](#)

plotLCTM, [11](#)

relative_entropy, [11](#)

residualplot_step1, [12](#)

sastraj_to_lctm, [13](#)