Package ‘LKT’

February 21, 2022

Title Logistic Knowledge Tracing
Version 1.1.0
Description Computes Logistic Knowledge Tracing (‘LKT’) which is a general method for tracking human learning in an educational software system. Please see Pavlik, Eglington, and Harrell-Williams (2021) <https://ieeexplore.ieee.org/document/9616435>. ‘LKT’ is a method to compute features of student data that are used as predictors of subsequent performance. ‘LKT’ allows great flexibility in the choice of predictive components and features computed for these predictive components. The system is built on top of ‘LiblineaR’, which enables extremely fast solutions compared to base glm() in R.

License GPL-3
Encoding UTF-8
LazyData true
VignetteBuilder knitr
RoxygenNote 7.1.2
Depends R (>= 3.5.0), SparseM (>= 1.78), Matrix, methods, data.table (>= 1.13.2), LiblineaR (>= 2.10-8)
Imports glmnet (>= 4.0-2), glmnetUtils (>= 1.1.8), lme4 (>= 1.1-23)
Suggests rmarkdown, pROC (>= 1.16.2), knitr, caret, utils
NeedsCompilation no
Author Philip I. Pavlik Jr. [aut, ctb, cre]
   (<https://orcid.org/0000-0001-6467-9452>),
   Luke G. Eglington [aut, ctb] (<https://orcid.org/0000-0002-8432-9203>)
Maintainer Philip I. Pavlik Jr. <imrryr@gmail.com>
Repository CRAN
Date/Publication 2022-02-21 13:30:16 UTC

R topics documented:

  computefeatures .................................................. 2
  computeSpacingPredictors ........................................ 3
computefeatures

countOutcome ......................................................... 3
LKT ................................................................. 4
samplelkt ............................................................. 6
smallSet ............................................................... 7

Index 8

Description

Compute feature describing prior practice effect.

Usage

computefeatures(data, feat, par1, par2, index, index2, par3, par4, par5, fcomp)

Arguments

data copy of main data frame.
feat is the feature to be computed.
par1 nonlinear parameters used for nonlinear features.
par2 nonlinear parameters used for nonlinear features.
index a student by component levels index
index2 a component levels index
par3 nonlinear parameters used for nonlinear features.
par4 nonlinear parameters used for nonlinear features.
par5 nonlinear parameters used for nonlinear features.
fcomp the component name.

Value

a vector suitable for regression input.
computeSpacingPredictors

desc

Description
Compute repetition spacing time based features from input data CF..Time. and/or CF..reltime. which will be automatically computed from Duration..sec. if not present themselves.

Usage
computeSpacingPredictors(data, KCs)

Arguments
- data: is a dataset with Anon.Student.Id and CF..ansbin.
- KCs: are the components for which spaced features will be specified in LKT

Value
data which is the same frame with the added spacing relevant columns.

countOutcome

desc

Description
Compute the prior sum of the response appearing in the outcome column for the index

Usage
countOutcome(data, index, response)

Arguments
- data: the dataset to compute an outcome vector for
- index: the subsets to count over
- response: the actually response value being counted

Value
the vector of the lagged cumulative sum.
Description

Compute a logistic regression model of learning for input data.

Usage

LKT(
  data,
  components,
  features,
  connectors = rep("+", length(components)),
  fixedpars = NA,
  seedpars = NA,
  covariates = NA,
  dualfit = FALSE,
  interc = FALSE,
  cv = FALSE,
  elastic = FALSE,
  verbose = TRUE,
  epsilon = 1e-04,
  cost = 512,
  type = 0,
  maketimes = FALSE,
  bias = 0
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>A dataset with Anon.Student.Id and CF..ansbin.</td>
</tr>
<tr>
<td>components</td>
<td>A vector of factors that can be used to compute each features for each subject.</td>
</tr>
<tr>
<td>features</td>
<td>A vector methods to use to compute a feature for the component.</td>
</tr>
<tr>
<td>connectors</td>
<td>A vector of the characters used for the formula connections, i.e., &quot;+&quot;, &quot;:&quot;, or &quot;*&quot;, with default &quot;+&quot; when not provided</td>
</tr>
<tr>
<td>fixedpars</td>
<td>A vector of parameters for all features+components.</td>
</tr>
<tr>
<td>seedpars</td>
<td>A vector of parameters for all features+components to seed non-linear parameter search.</td>
</tr>
<tr>
<td>covariates</td>
<td>A list of components that interacts with component by feature in the main specification.</td>
</tr>
<tr>
<td>dualfit</td>
<td>TRUE or FALSE, fit a simple latency using logit. Requires Duration..sec. column in data.</td>
</tr>
<tr>
<td>interc</td>
<td>TRUE or FALSE, include a global intercept.</td>
</tr>
</tbody>
</table>
cv
TRUE or FALSE, if TRUE runs N-fold cv. Requires premade column named 'fold' with integers denoting the N folds
elastic
glmnet, cv.glmnet, cva.glmnet or FALSE.
verbose
provides more output in some cases.
epsilon
passed to LiblineaR
cost
passed to LiblineaR
type
passed to LiblineaR
maketimes
Boolean indicating whether to create time based features (or may be precomputed)
bias
passed to LiblineaR

Value
list of values "model", "coefs", "r2", "prediction", "nullmodel", "latencymodel", "optimizedpars", "subjectrmse", "newdata", and "loglike"

Examples
```R
temp <- samplelkt
temp$CF..ansbin.<-ifelse(temp$Outcome=="CORRECT",1,ifelse(temp$Outcome=="INCORRECT",0,-1))
temp <- data.table::setDT(temp)
temp <- computeSpacingPredictors(temp, "KC..Default.")
temp <- temp[temp$CF..ansbin==0 | temp$CF..ansbin==1,]
temp$KC..Default.<-substr(temp$KC..Default.,1,10)
modelob <- LKT(
data = temp, interc=TRUE,
components = c("Anon.Student.Id", "KC..Default.", "KC..Default."),
features = c("logitdec", "logitdec", "lineafm"),
fixedpars = c(.9, .85)
)
print(modelob$coefs)
print(modelob$loglik)
```
```R
modelob <- LKT(
data = temp, interc=TRUE,
components = c("Anon.Student.Id", "KC..Default.", "KC..Default."),
features = c("intercept", "logitdec", "lineafm"),
seedpars = c(.9, .85)
)
print(modelob$coefs)
print(modelob$loglik)
```
```R
modelob <- LKT(
data = temp, interc=TRUE,
connectors = c("+","x","+")
components = c("Anon.Student.Id", "KC..Default.", "KC..Default."),
features = c("intercept", "logitdec", "lineafm"),
fixedpars = c(.9, .85)
)
```
## Description

A dataset containing a small sample of participants in a memory experiment.
smallSet

Usage

sample1kt

Format

A data frame with 2074 rows and many variables:

Anon.Student.Id unique identifier for each student
Duration..sec. unique identifier for each student
KC..Default. unique identifier for each student
Outcome unique identifier for each student ...

Source

https://datashop.memphis.edu/index.jsp

smallSet(data, nSub)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>Dataframe of student data</td>
</tr>
<tr>
<td>nSub</td>
<td>Number of students</td>
</tr>
</tbody>
</table>
Index

* datasets
  samplelkt, 6
  computefeatures, 2
  computeSpacingPredictors, 3
  countOutcome, 3

LKT, 4

samplelkt, 6
smallSet, 7