

# Package ‘LKT’

May 7, 2026

**Title** Logistic Knowledge Tracing

**Version** 1.7.0

**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 Harrel-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** 7.2.3

**Depends** R (>= 3.5.0), SparseM (>= 1.83), methods, Matrix, data.table (>= 1.13.2), LiblineaR (>= 2.10-8)

**Imports** glmnet (>= 4.0-2), glmnetUtils (>= 1.1.8), lme4 (>= 1.1-23), cluster (>= 2.1.3), pROC (>= 1.16.2), crayon, HDInterval (>= 0.2.2)

**Suggests** rmarkdown, knitr, utils, caret, ggplot2

**NeedsCompilation** no

**Author** Philip I. Pavlik Jr. [aut, ctb, cre] (ORCID:  
<<https://orcid.org/0000-0001-6467-9452>>),  
Luke G. Eglington [aut, ctb] (ORCID:  
<<https://orcid.org/0000-0002-8432-9203>>)

**Maintainer** Philip I. Pavlik Jr. <imrryr@gmail.com>

**Repository** CRAN

**Date/Publication** 2024-07-01 22:40:12 UTC

## Contents

buildLKTModel . . . . .	2
computefeatures . . . . .	4
computeSpacingPredictors . . . . .	4
countOutcomeold . . . . .	5
largerawsample . . . . .	5
LASSOLKTData . . . . .	6
LASSOLKTModel . . . . .	7
LKT . . . . .	8
LKT_HDI . . . . .	10
predict_lkt . . . . .	11
samplelkt . . . . .	12
smallSet . . . . .	12
ViewExcel . . . . .	13
<b>Index</b>	<b>14</b>

---

buildLKTModel	<i>buildLKTModel</i>
---------------	----------------------

---

## Description

Forward and backwards stepwise search for a set of features and components with tracking of nonlinear parameters.

## Usage

```
buildLKTModel(
  data,
  usefolds = NA,
  allcomponents,
  allfeatures,
  currentcomponents = c(),
  specialcomponents = c(),
  specialfeatures = c(),
  forv,
  bacv,
  preset = NA,
  presetint = T,
  currentfeatures = c(),
  verbose = FALSE,
  currentfixedpars = c(),
  maxitv = 10,
  interc = FALSE,
  forward = TRUE,
  backward = TRUE,
  metric = "BIC",
```

```

  removefeat = c(),
  removecomp = c()
)

```

### Arguments

<code>data</code>	is a dataset with <code>Anon.Student.Id</code> and <code>CF.ansbin</code> .
<code>usefolds</code>	Numeric Vector   Specifies the folds for model fitting in LKT; the features are still calculated across all folds to compute test fold fit externally
<code>allcomponents</code>	is search space for LKT components
<code>allfeatures</code>	is search space for LKT features
<code>currentcomponents</code>	components to start search from
<code>specialcomponents</code>	add special components (not crossed with features, only paired with special features 1 for 1)
<code>specialfeatures</code>	features for each special component (not crossed during search)
<code>forv</code>	the minimum amount of improvement needed for the addition of a new term
<code>bacv</code>	the maximum amount of loss for a term to be removed
<code>preset</code>	One of "static", "AFM", "PFA", "advanced", "AFMLLTM", "PFALLTM", "advancedLLTM"
<code>presetint</code>	should the intercepts be included for preset components
<code>currentfeatures</code>	features to start search from
<code>verbose</code>	passed to LKT
<code>currentfixedpars</code>	used for current features as an option to start
<code>maxitv</code>	passed to LKT
<code>interc</code>	passed to LKT
<code>forward</code>	TRUE or FALSE
<code>backward</code>	TRUE or FALSE
<code>metric</code>	One of "BIC", "AUC", "AIC", and "RMSE"
<code>removefeat</code>	Character Vector   Excludes specified features from the test list.
<code>removecomp</code>	Character Vector   Excludes specified components from the test list.

### Value

list of values "tracetable" and "currentfit"

---

computefeatures	<i>computefeatures</i>
-----------------	------------------------

---

**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	<i>computeSpacingPredictors</i>
--------------------------	---------------------------------

---

**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.

---

countOutcomeold	<i>countOutcome</i>
-----------------	---------------------

---

**Description**

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

**Usage**

```
countOutcomeold(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.

---

largerawsample	<i>Trial sequences for practice participants.</i>
----------------	---

---

**Description**

A dataset containing a raw sample from the Memphis DataShop.

**Usage**

```
largerawsample
```

**Format**

A data frame please see the DataShop for more info.  
 It has many columns.

**Source**

<https://psl1datashop.web.cmu.edu/Export?datasetId=5513>

---

LASSOLKTData

*LASSOLKTData*


---

### Description

Forward and backwards stepwise search for a set of features and components with tracking of nonlinear parameters.

### Usage

```
LASSOLKTData(
  data,
  gridpars,
  allcomponents,
  allfeatures,
  preset = NA,
  presetint = T,
  specialcomponents = c(),
  specialfeatures = c(),
  specialpars = c(),
  removefeat = c(),
  removecomp = c()
)
```

### Arguments

data	is a dataset with Anon.Student.Id and CF.ansbin.
gridpars	a vector of parameters to create each feature at
allcomponents	is search space for LKT components
allfeatures	is search space for LKT features
preset	One of "static", "AFM", "PFA", "advanced", "AFMLLTM", "PFALLTM", "advancedLLTM"
presetint	should the intercepts be included for preset components
specialcomponents	add special components (not crossed with features, only paired with special features 1 for 1)
specialfeatures	features for each special component (not crossed during search)
specialpars	parameters for the special features (if needed)
removefeat	Character Vector   Excludes specified features from the test list.
removecomp	Character Vector   Excludes specified components from the test list.

### Value

data which is the same frame with the added spacing relevant columns.  
list of values "tracetable" and "currentfit"

LASSOLKTModel

*LASSOLKTModel***Description**

runs LASSO search on the data

**Usage**

```
LASSOLKTModel(
  data,
  gridpars,
  allcomponents,
  preset = NA,
  presetint = T,
  allfeatures,
  specialcomponents = c(),
  specialfeatures = c(),
  specialpars = c(),
  target_n,
  removefeat = c(),
  removecomp = c(),
  test_fold = 1
)
```

**Arguments**

<code>data</code>	is a dataset with Anon.Student.Id and CF.ansbin.
<code>gridpars</code>	a vector of parameters to create each feature at
<code>allcomponents</code>	is search space for LKT components
<code>preset</code>	One of "static", "AFM", "PFA", "advanced", "AFMLLTM", "PFALLTM", "advancedLLTM"
<code>presetint</code>	should the intercepts be included for preset components
<code>allfeatures</code>	is search space for LKT features
<code>specialcomponents</code>	add special components (not crossed with features, only paired with special features 1 for 1)
<code>specialfeatures</code>	features for each special component (not crossed during search)
<code>specialpars</code>	parameters for the special features (if needed)
<code>target_n</code>	chosen number of features in model
<code>removefeat</code>	Character Vector   Excludes specified features from the test list.
<code>removecomp</code>	Character Vector   Excludes specified components from the test list.
<code>test_fold</code>	the fold that the chosen LASSO model will be tested on

**Value**

list of matrices and values "train\_x", "train\_y", "test\_x", "test\_y", "fit", "target\_auc", "target\_rmse", "n\_features", "auc\_lambda", "preds"

---

LKT

*LKT*


---

**Description**

Compute a logistic regression model of learning for input data.

**Usage**

```
LKT(
  data,
  usefolds = NA,
  components,
  features,
  fixedpars = NA,
  seedpars = NA,
  interacts = NA,
  curvefeats = NA,
  dualfit = FALSE,
  interc = FALSE,
  verbose = TRUE,
  epsilon = 1e-04,
  cost = 512,
  lowb = 1e-05,
  highb = 0.99999,
  type = 0,
  maketimes = FALSE,
  bias = 0,
  maxitv = 100,
  factrv = 1e+12,
  nosolve = FALSE,
  autoKC = rep(0, length(components)),
  autoKCcont = rep("NA", length(components)),
  connectors = rep("+", max(1, length(components) - 1))
)
```

**Arguments**

data	A dataset with Anon.Student.Id and CF..ansbin.
usefolds	Numeric Vector   Specifies the folds for model fitting in LKT; the features are still calculated across all folds to compute test fold fit externally
components	A vector of factors that can be used to compute each features for each subject.

features	a vector methods to use to compute a feature for the component.
fixedpars	a vector of parameters for all features+components.
seedpars	a vector of parameters for all features+components to seed non-linear parameter search.
interacts	A list of components that interacts with component by feature in the main specification.
curvefeats	vector of columns to use with "diff" functions
dualfit	TRUE or FALSE, fit a simple latency using logit. Requires Duration..sec. column in data.
interc	TRUE or FALSE, include a global intercept.
verbose	provides more output in some cases.
epsilon	passed to LiblinearR
cost	passed to LiblinearR
lowb	lower bound for non-linear optimizations
highb	upper bound for non-linear optimizations
type	passed to LiblinearR
maketimes	Boolean indicating whether to create time based features (or may be precomputed)
bias	passed to LiblinearR
maxitv	passed to nonlinear optimization a maxit control
factrv	controls the optim() function
nosolve	causes the function to return a sparse data matrix of the features, rather than a solution
autoKC	a vector to indicate whether to use autoKC for the component (0) or the k for the numebr of clusters
autoKCcont	a vector of text strings set to "rand" for component to make autoKC assignment to cluster is randomized (for comaprison)
connectors	a vector if linear equation R operators including +, * and :

### Value

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

---

LKT\_HDI

*LKT\_HDI*


---

### Description

Bootstrap credibility intervals to aid in interpreting coefficients.

### Usage

```
LKT_HDI(
  dat,
  n_boot,
  n_students,
  comps,
  feats,
  conns = rep("+", max(1, length(comps) - 1)),
  ints = NA,
  fixeds,
  get_hdi = TRUE,
  cred_mass = 0.95
)
```

### Arguments

<code>dat</code>	Dataframe
<code>n_boot</code>	Number of subsamples to fit
<code>n_students</code>	Number of students per subsample
<code>comps</code>	Components in model
<code>feats</code>	Features in model
<code>conns</code>	R notation for linear equation connectors in model
<code>ints</code>	Interacts in model
<code>fixeds</code>	Fixed parameters in model
<code>get_hdi</code>	Boolean to decide if generating HDI per coefficient
<code>cred_mass</code>	Credibility mass parameter to decide width of HDI

### Value

List of values "par\_reps", "mod\_full", "coef\_hdi"

---

predict\_lkt                      *Predict for LKT Models*

---

### Description

Generates predictions and evaluates logistic regression models tailored for learning data, specifically designed for Logistic Knowledge Tracing (LKT) models. This function provides flexibility in returning either just the predicted probabilities or both the predictions and key evaluation statistics.

### Usage

```
predict_lkt(  
  modelob,  
  data,  
  fold = NULL,  
  return_stats = FALSE,  
  min_pred_limit = 1e-05,  
  max_pred_limit = 0.99999  
)
```

### Arguments

modelob	An LKT model object containing necessary model coefficients and predictors for generating predictions.
data	A dataset including predictor variables, the outcome variable <code>CF..ansbin.</code> , and fold information.
fold	Optional. Numeric vector specifying which folds to include for prediction. If NULL or empty, uses all data.
return_stats	Logical. If TRUE, returns both predictions and evaluation statistics (Log-Likelihood, AUC, RMSE, R <sup>2</sup> ). If FALSE, returns only the predictions.
min_pred_limit	Minimum prediction limit. Default is 0.00001.
max_pred_limit	Maximum prediction limit. Default is 0.99999.

### Value

If `return_stats` is FALSE, returns a list containing:

- `predictions`: The predicted probabilities for each observation in the specified fold(s).

If `return_stats` is TRUE, returns a list containing:

- `predictions`: The predicted probabilities for each observation in the specified fold(s).
- `LL`: Log-Likelihood of the model given the actual outcomes.
- `AUC`: Area Under the ROC Curve.
- `RMSE`: Root Mean Squared Error.
- `R2`: R-squared value, indicating the proportion of variance explained by the model.

---

samplelkt	<i>Trial sequences for practice participants.</i>
-----------	---

---

**Description**

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

**Usage**

```
samplelkt
```

**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://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=5508>

---

smallSet	<i>smallSet</i>
----------	-----------------

---

**Description**

smallSet

**Usage**

```
smallSet(data, nSub)
```

**Arguments**

data            Dataframe of student data

nSub            Number of students

---

ViewExcel

*ViewExcel*

---

**Description**

ViewExcel

**Usage**

```
ViewExcel(df = .Last.value, file = tempfile(fileext = ".csv"))
```

**Arguments**

df	Dataframe
file	name of the Excel file

# Index

## \* datasets

- largerawsample, [5](#)
- samplelkt, [12](#)

buildLKTModel, [2](#)

computefeatures, [4](#)

computeSpacingPredictors, [4](#)

countOutcomeold, [5](#)

largerawsample, [5](#)

LASSOLKTData, [6](#)

LASSOLKTModel, [7](#)

LKT, [8](#)

LKT\_HDI, [10](#)

predict\_lkt, [11](#)

samplelkt, [12](#)

smallSet, [12](#)

ViewExcel, [13](#)