Package 'BKT'

January 30, 2025

Title Bayesian Knowledge Tracing Model

Version 0.0.1

Description

Fitting, cross-validating, and predicting with Bayesian Knowledge Tracing (BKT) models. It is designed for analyzing educational datasets to trace student knowledge over time. The package includes functions for fitting BKT models, evaluating their performance using various metrics, and making predictions on new data. It provides the same functionality as the Python package pyBKT authored by Bulut et al. (2023, <doi:10.3390/psych5030050>) and pyBKT can be accessed via the site at <https://github.com/CAHLR/pyBKT>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports RCurl, parallel, methods, stats, utils,

Suggests testthat (>= 3.0.0)

NeedsCompilation no

Author Yuhao Yuan [aut, cre],
Biying Zhou [aut],
Feng Ji [aut]

Maintainer Yuhao Yuan <yuanyuhaoapply@163.com>

Repository CRAN

Date/Publication 2025-01-30 16:50:06 UTC

Contents

Index   **13**

---

    bkt   *Bayesian Knowledge Tracing*

Description

Create a BKT (Bayesian Knowledge Tracing) model object with initial parameters. This function constructs a BKT model by taking in various parameters such as parallelization options, number of fits, random seed, and other model-specific settings. These parameters can later be modified during the fitting or cross-validation process.

Usage

```
bkt(
  parallel = TRUE,
  num_fits = 5,
  folds = 5,
  seed = sample(1:1e+08, 1),
  model_type = rep(FALSE, 4),
  forgets = FALSE,
  fixed = NULL,
  defaults = NULL,
  ...
)
```

Arguments

parallel	Logical. Indicates whether to use parallel computation. If set to `TRUE`, multi-threading will be used to speed up model training.
num_fits	Integer. Number of fit iterations. The best model is selected from the total iterations.
folds	Integer. Number of folds used for cross-validation. This parameter is used during cross-validation to divide the data into parts.
seed	Numeric. Seed for the random number generator, which ensures reproducibility of results.
model_type	Logical vector. Specifies model variants to use. There are four possible variants: 'multilearn', 'multiprior', 'multipair', and 'multigs'. Each corresponds to a different modeling strategy.
forgets	Logical. Whether to include a forgetting factor in the model. If set to `TRUE`, the model will account for the possibility that learners may forget knowledge.

fixed	List. A nested list specifying which parameters to fix for specific skills during model fitting. Each skill can have certain parameters, such as "guesses" and "slips", set to TRUE (to fix) or FALSE (to let them vary). For example: list("skill_name" = list("guesses" = TRUE, "slips" = TRUE)).
defaults	List. The defaults parameter is a list that functions as a query dictionary. It is used to map column names in the data to the expected variables in the model. This helps ensure that the model can work with different datasets that may have varying column names.
...	Other parameters.

Value

A BKT model object, which can be used by other functions such as fitting the model, cross-validation, or making predictions.

Examples

```
model <- bkt(seed = 42, parallel = FALSE, num_fits = 1)
```

---

crossvalidate	*Cross Validation*

Description

Perform cross-validation on a BKT (Bayesian Knowledge Tracing) model. This function trains and evaluates the BKT model using cross-validation. It splits the dataset into training and validation sets, trains the model on the training data, and evaluates it on the validation data according to a specified metric.

Usage

```
crossvalidate(
  object,
  data = NULL,
  data_path = NULL,
  metric = rmse,
  parallel = FALSE,
  seed = NULL,
  num_fits = 1,
  folds = 5,
  forgets = FALSE,
  fixed = NULL,
  model_type = NULL,
  ...
)
```

Arguments

`object`	A BKT model object. The model to be cross-validated.
`data`	Data frame. The dataset to be used for cross-validation. If `data` is not provided, `data_path` should be used to load the dataset from a file.
`data_path`	Character. The file path to the dataset. This will be used if `data` is not provided.
`metric`	Function. The metric function used to evaluate model performance.
`parallel`	Logical. Indicates whether to use parallel computation. If set to `TRUE`, multi-threading will be used to speed up model training.
`seed`	Numeric. Seed for the random number generator, which ensures reproducibility of results.
`num_fits`	Integer. Number of fit iterations. The best model is selected from the total iterations.
`folds`	Integer. Number of folds used for cross-validation. This parameter is used during cross-validation to divide the data into parts.
`forgets`	Logical. Whether to include a forgetting factor in the model. If set to `TRUE`, the model will account for the possibility that learners may forget knowledge.
`fixed`	List. A nested list specifying which parameters to fix for specific skills during model fitting. Each skill can have certain parameters, such as "guesses" and "slips", set to `TRUE` (to fix) or `FALSE` (to let them vary). For example: `list("skill_name" = list("guesses" = TRUE, "slips" = TRUE))`.
`model_type`	Logical vector. Specifies model variants to use. There are four possible variants: 'multilearn', 'multiprior', 'multipair', and 'multigs'. Each corresponds to a different modeling strategy.
`...`	Other parameters.

Value

A list containing the cross-validation results, including the average performance metric and any other relevant details from the validation process.

Examples

```
model <- bkt(seed = 42, parallel = TRUE, num_fits = 5)
cv_results <- crossvalidate(model, data_path = "ct.csv", folds = 5)
print(cv_results)
```

---

evaluate *Evaluate*

Description

Evaluate a BKT (Bayesian Knowledge Tracing) model using a specified metric. This function evaluates a fitted BKT model on a given dataset using a chosen performance metric. It takes either a data frame or a file path to the data and returns the evaluation result based on the specified metric (e.g., RMSE or accuracy).

Usage

```
evaluate(object, data = NULL, data_path = NULL, metric = rmse)
```

Arguments

object	A fitted BKT model object. This is the model to be evaluated.
data	Data frame. The dataset on which the model will be evaluated. If data is not provided, the function will attempt to load the dataset from the file specified by data_path.
data_path	Character. The file path to the dataset for evaluation. This will be used if data is not provided.
metric	Function or Function List. The evaluation metric used to assess the model performance. (Root Mean Square Error), but other metrics can also be specified.

Value

Numeric or List. The result of the evaluation based on the specified metric(s). For example, if rmse is used, the function will return the root mean square error for the model on the dataset.

Examples

```
model <- bkt(seed = 42, parallel = TRUE, num_fits = 5)
result <- fit(model, data_path = "ct.csv", skills = "Plot non-terminating improper fraction")
eval_result <- evaluate(result, data_path = "ct_test.csv", metric = rmse)
print(eval_result)
```

---

fetch_dataset   *Fetch a dataset*

Description

Fetch a dataset from an online source. This function downloads a dataset from a provided URL and saves it to a specified location on the local system. The dataset must be publicly accessible, without requiring any password or authentication. It can then be used for further analysis or modeling.

Usage

```
fetch_dataset(object, link, loc)
```

Arguments

object	A BKT model object. The model can use the fetched dataset for fitting or other tasks.
link	Character. The URL where the dataset is located. This must be a publicly accessible URL.
loc	Character. The local file path where the dataset will be saved. The dataset will be stored at this location after download.

Value

None. The function downloads the data file to the specified location.

Examples

```
model <- bkt()
fetch_dataset(model, "http://example.com/dataset.csv", "data.csv")
```

---

fit   *fit bkt model*

Description

Fit a BKT (Bayesian Knowledge Tracing) model. This function fits the BKT model using the provided data and various options, such as skill filtering, forget model, and parallelization. The function uses the model object created by bkt() and fits the data according to the specified parameters.

Usage

```
fit(
  object,
  data_path = NULL,
  data = NULL,
  parallel = FALSE,
  seed = NULL,
  num_fits = 1,
  forgets = FALSE,
  fixed = NULL,
  model_type = NULL,
  ...
)
```

Arguments

`object`	A BKT model object. The model to be cross-validated.
`data_path`	Character. The file path to the dataset. This will be used if `data` is not provided.
`data`	Data frame. The dataset to be used for cross-validation. If `data` is not provided, `data_path` should be used to load the dataset from a file.
`parallel`	Logical. Indicates whether to use parallel computation. If set to `TRUE`, multi-threading will be used to speed up model training.
`seed`	Numeric. Seed for the random number generator, which ensures reproducibility of results.
`num_fits`	Integer. Number of fit iterations. The best model is selected from the total iterations.
`forgets`	Logical. Whether to include a forgetting factor in the model. If set to `TRUE`, the model will account for the possibility that learners may forget knowledge.
`fixed`	List. A nested list specifying which parameters to fix for specific skills during model fitting. Each skill can have certain parameters, such as "guesses" and "slips", set to `TRUE` (to fix) or `FALSE` (to let them vary). For example: `list("skill_name" = list("guesses" = TRUE, "slips" = TRUE))`.
`model_type`	Logical vector. Specifies model variants to use. There are four possible variants: 'multilearn', 'multiprior', 'multipair', and 'multigs'. Each corresponds to a different modeling strategy.
`...`	Other parameters.

Value

A fitted BKT model object, which can be used for predictions, cross-validation, or parameter analysis.

Examples

```
model <- bkt(seed = 42, parallel = FALSE, num_fits = 1)
result <- fit(
```

```
  model,
  data_path = "data.csv"
)
```

---

load_model   *Load*

Description

Load a BKT model from a file. This function loads a previously saved BKT model from an RDS file. The model attributes are restored into the provided model object, allowing it to be used for further analysis or predictions.

Usage

```
load_model(model, loc)
```

Arguments

model   A BKT model object into which the saved model's attributes will be loaded.

loc   Character. The file path from which the model will be loaded, typically an `.rds` file.

Value

The updated BKT model object with the restored attributes from the saved model.

Examples

```
model <- bkt(seed = 42)
loaded_model <- load_model(model, "bkt_model.rds")
```

---

params   *Extract Parameters from BKT model*

Description

Extract fitted parameters from a BKT model. This function retrieves the parameters from a fitted BKT model object. The parameters include model-specific values such as "learns", "guesses", "slips", and "forgets". These parameters are returned in a format that is easy to print or manipulate for further analysis.

Usage

```
params(object)
```

Arguments

object   A fitted BKT model object. The model should have been previously fitted using
                the `fit()` function, otherwise no parameters will be available.

Value

A data frame containing the fitted model parameters. The data frame will typically include columns
such as 'learns', 'guesses', 'slips', and other model-specific values.

Examples

```
model <- bkt(seed = 42, parallel = TRUE, num_fits = 5)
result <- fit(model, data_path = "data.csv", skills = "skill name")
params_df <- params(result)
print(params_df)
```

---

predict_bkt   *Predict*

Description

Predict outcomes using a fitted BKT model. This function uses a trained Bayesian Knowledge
Tracing (BKT) model to make predictions on new data. The predictions include both the likeli-
hood of a correct response (`correct_predictions`) and the estimated hidden state of the learner's
knowledge (`state_predictions`).

Usage

```
predict_bkt(model, data_path = NULL, data = NULL)
```

Arguments

model   A trained BKT model object. The model must have been previously fitted using
                the `fit()` function. If the model is not fitted, an error will be raised.

data_path   Character. The file path to the dataset on which predictions will be made. If this
                is provided, the function will read data from the file.

data   Data frame. A pre-loaded dataset to be used for predictions. This can be used
                instead of specifying `data_path`.

Value

A data frame containing the original data with two additional columns: `correct_predictions` and
`state_predictions`.

Examples

```
model <- bkt(seed = 42)
fit_model <- fit(model, data_path = "ct.csv")
predictions <- predict_bkt(fit_model, data_path = "ct_test.csv")
head(predictions)
```

---

save_model   *Save*

Description

Save a BKT model to a file. This function saves a trained BKT model to a specified file location. The model is stored as an RDS file, which can be loaded back into R using the load_model() function.

Usage

```
save_model(model, loc)
```

Arguments

model	A trained BKT model object to be saved.
loc	Character. The file path where the model will be saved, typically with an .rds extension.

Value

None. The function saves the model to the specified location.

Examples

```
model <- bkt(seed = 42)
fit_model <- fit(model, data_path = "ct.csv")
save_model(fit_model, "bkt_model.rds")
```

---

set_coef   *Set Coefficients for BKT Model*

Description

This function sets or initializes the parameters of a Bayesian Knowledge Tracing (BKT) model. The user can manually specify the values for different parameters associated with specific skills.

Usage

```
set_coef(object, values)
```

Arguments

object   An object of the BKT model. This is the model for which the parameters will be set or initialized.

values   A list containing the skill names and their corresponding BKT parameters. Each skill should have its own list of parameters. The parameters can include 'prior', 'learns', 'forgets', 'guesses', and 'slips'. Example structure: `list("skill_name" = list("learns" = ..., "guesses" = ...))`.

Details

This function allows users to manually specify or update the parameters of a BKT model for different skills. The values should be provided as a named list, with each skill having its own sublist of BKT parameters. The function performs checks to ensure that the provided parameters are valid in terms of type, length, and existence.

Value

The updated BKT model object with the newly set coefficients.

Examples

```
# Initialize a BKT model
model <- bkt(seed = 42)

# Set custom parameters for a specific skill
model <- set_coef(model, list(
  "Plot non-terminating improper fraction" = list("prior" = 0.5, "learns" = 0.2)
))

# Fit the model with fixed parameters
result <- fit(model,
  forgets = TRUE,
  data_path = "ct.csv",
  skills = "Plot non-terminating improper fraction",
  fixed = list("Plot non-terminating improper fraction" = list("prior" = TRUE))
```

)

Index