Package 'ConvertPar'

May 7, 2026

Type Package

Title Estimating IRT Parameters via Machine Learning Algorithms

Version 0.1

Maintainer Eda Akdogdu Yildiz <akdogdueda@gmail.com>

Description A tool to estimate IRT item parameters (2 PL) using CTT-based item statistics from small samples via artificial neural networks and regression trees.

Imports neuralnet, mirt, RWeka, stats

License GPL (>= 3)

Encoding UTF-8

NeedsCompilation no

Author Eda Akdogdu Yildiz [aut, cre] (ORCID: <https://orcid.org/0000-0003-4374-4379>), Huseyin Yıldız [aut] (ORCID: <https://orcid.org/0000-0003-2387-263X>), Kubra Atalay Kabasakal [aut] (ORCID: <https://orcid.org/0000-0002-3580-5568>)

Repository CRAN

Date/Publication 2022-03-14 19:20:05 UTC

Contents

conv.ann *Estimating IRT Item Parameters with Small Samples via Artificial Neural Networks*

Description

This function can be used to estimate IRT item parameters (2 PL) using CTT-based item statistics from small samples via artificial neural networks.

Usage

```
conv.ann(small.data, train.data, model="2PL",layers=1,learningrate=NULL,treshold=0.01)
```

Arguments

small.data	matrix or data frame: contains small sample dichotomous participant response matrix.
train.data	matrix or data frame: contains a dichotomous response matrix to use training of ANN model. This matrix should be contain as much as possible participants for more accurate estimations.The "gen.data" function can be used to obtain a simulative response matrix.
model	string: option for desired IRT model. 'Rasch' or '2PL' ('2PL' is default)
layers	vector: a vector of integers specifying the number of hidden neurons (vertices) in each layer.
learningrate	numeric: a numeric value specifying the learning rate.
treshold	numeric: a numeric value specifying the threshold for the partial derivatives of the error function as stopping criteria.

Value

This function returns a list including following:

- a matrix: Predicted IRT Parameters
- a matrix: Item Parameters of Training Data

Examples

```
## Genarating item and ability parameters (1000 participants, 100 items)

a <- rlnorm(100,0,0.3)
b <- rnorm(100,0,1)
responses <- matrix(NA, nrow=1000, ncol=100)
theta <- rnorm(1000, 0,1)

### Defining Response Function (2 PL)

pij <- function(a,b,theta) {
```

```
      1/(1+exp(-1*a*(theta-b)))
    }

### Creating Response Matrix and column names.

  for( i in 1:1000 ) {
   for( j in 1:100 ) {
     responses[i,j]<-ifelse(pij(a=a[j], b=b[j], theta[i]) < runif(1) , 0 ,1)

   }
  }

  names<-paste("i",1:ncol(responses),sep = "_")

  colnames(responses)<-names
  train<-as.data.frame(responses)

  small.index<-sample(1:nrow(train),100,replace=FALSE)

  small<-train[small.index,]


  ### Conducting Function

  conv.ann(small.data=small, train.data=train, model="2PL",layers=c(2,2),
  learningrate=NULL,treshold=0.01)
```

---

conv.rt *Estimating IRT Item Parameters with Small Samples via Regression Trees*

Description

This function can be used to estimate IRT item parameters (2 PL) using CTT-based item statistics from small samples via Regression Trees.

Usage

```
conv.rt(small.data, train.data, model="2PL",pruned=TRUE,min.inst=10)
```

Arguments

small.data	matrix or data frame: contains small sample dichotomous participant response matrix.
train.data	matrix or data frame: contains a dichotomous response matrix to use training of ANN model. This matrix should be contain as much as possible participants for more accurate estimations.The "gen.data" function can be used to obtain a simulative response matrix.

model	string: option for desired IRT model. 'Rasch' or '2PL' ('2PL' is default)
pruned	a logical: Use unpruned tree/rules. Default is TRUE
min.inst	numeric: Minimum number of items per leaf (Default 10).

Value

This function returns a list including following:

- a matrix: Predicted IRT Parameters
- a matrix: Item Parameters of Training Data
- a list: Tree Models and Regression Equations

Examples

```
## Genarating item and ability parameters (1000 participants, 100 items)

a <- rlnorm(100,0,0.3)
b <- rnorm(100,0,1)
responses <- matrix(NA, nrow=1000, ncol=100)
theta <- rnorm(1000, 0,1)

### Defining Response Function (2 PL)

pij <- function(a,b,theta) {
    1/(1+exp(-1*a*(theta-b)))
  }

### Creating Response Matrix and column names.

 for( i in 1:1000 ) {
  for( j in 1:100 ) {
    responses[i,j]<-ifelse(pij(a=a[j], b=b[j], theta[i]) < runif(1) , 0 ,1)

  }
}

names<-paste("i",1:ncol(responses),sep = "_")

colnames(responses)<-names
train<-as.data.frame(responses)

small.index<-sample(1:nrow(train),100,replace=FALSE)

small<-train[small.index,]


### Conducting Function

conv.rt(small.data=small,
train.data=train,
model="2PL",
```

```
pruned=TRUE,
min.inst=10)
```

---

gen.data *Generating Dichotomous Data Sets based on Logistic IRT Models (Rasch, 2PL, 3PL).*

Description

This function can be used for generating dichotomous response matrices based on Logistic IRT Models. Sample size, item number, parameter distributions can be specified.

Usage

```
gen.data(model="2PL",samplesize=1000,itemsize=100,
theta.mean=0,theta.sd=1, a.mean=0, a.sd=0.2,b.mean=0,
b.sd=1, c.min=0, c.max=0.25)
```

Arguments

model	string: option for desired IRT model. 'Rasch', '2PL' or '3PL' ('2PL' is default)
samplesize	numeric: Desired Sample size (Default 1000).
itemsize	numeric: Desired item number (Default 100).
theta.mean	numeric: mean value of theta normal distribution (Default 0).
theta.sd	numeric: standart deviation of theta normal distribution (Default 1).
a.mean	numeric: mean value of a parameters log normal distribution (Default 0).
a.sd	numeric: standart deviation of a parameters log normal distribution (Default 0.2).
b.mean	numeric: mean value of b parameters normal distribution (Default 0).
b.sd	numeric: standart deviation of b parameters normal distribution (Default 1).
c.min	numeric: minimum value of c parameters uniform distribution (Default 0).
c.max	numeric: maximum value of c parameters uniform distribution (Default 0.25).

Value

This function returns a a data frame containing simulated dichotomous response matrix.

Examples

```
gen.data(model="2PL",
samplesize=1000,
itemsize=100,
theta.mean=0,
theta.sd=1,
a.mean=0,
a.sd=0.2,
b.mean=0,
b.sd=1,
c.min=0,
c.max=0.25)
```

Index