Package 'IRTBEMM'

June 6, 2023

Type Package

Title Family of Bayesian EMM Algorithm for Item Response Models

Version 1.0.8

Date 2023-06-06

Description Applying the family of the Bayesian Expectation-Maximization-
Maximization (BEMM) algorithm to estimate: (1) Three parameter logistic (3PL) model pro-
posed by Birnbaum (1968, ISBN:9780201043105); (2) four parameter logistic (4PL) model pro-
posed by Barton & Lord (1981) <doi:10.1002/j.2333-8504.1981.tb01255.x>; (3) one parame-
ter logistic guessing (1PLG) and (4) one parameter logistic ability-based guessing (1PLAG) mod-
els proposed by San Martín et al (2006) <doi:10.1177/0146621605282773>. The BEMM fam-
ily includes (1) the BEMM algorithm for 3PL model pro-
posed by Guo & Zheng (2019) <doi:10.3389/fpsyg.2019.01175>; (2) the BEMM algo-
rithm for 1PLG model and (3) the BEMM algorithm for 1PLAG model pro-
posed by Guo, Wu, Zheng, & Chen (2021) <doi:10.1177/0146621621990761>; (4) the BEMM al-
gorithm for 4PL model pro-
posed by Zheng, Guo, & Kern (2021) <doi:10.1177/21582440211052556>; and (5) their maxi-
mum likelihood estimation versions pro-
posed by Zheng, Meng, Guo, & Liu (2018) <doi:10.3389/fpsyg.2017.02302>. Thus, both Bayesian modal es-
timates and maximum likelihood estimates are available.

License GPL (>= 2)

Depends R (>= 3.5.0)

NeedsCompilation no

Encoding UTF-8

Author Shaoyang Guo [aut, cre, cph],
Chanjin Zheng [aut],
Justin L Kern [aut]

Maintainer Shaoyang Guo <syguo1992@outlook.com>

Repository CRAN

Date/Publication 2023-06-06 12:20:05 UTC

R **topics documented:**

BEMM.1PLAG   *Calibrating 1PLAG model via Bayesian Expectation-Maximization-Maximization (BEMM) algorithm.*

Description

This function can estimate the item parameters of the 1PLAG model via Bayesian Expectation-Maximization-Maximization (BEMM) algorithm proposed by Guo, Wu, Zheng, & Wang (2018, April). Both Bayesan modal estimates and maximum likelihood estimates are available. In addition, the examinees' ability and a few model fits information can be also obtained through this function.

Usage

```
BEMM.1PLAG(data, PriorAlpha = c(-1.9, 1), PriorBeta = c(0, 4),
PriorGamma = c(-1.39, 0.25), InitialAlpha = NA,
InitialBeta = NA, InitialGamma = NA, Tol = 0.0001,
max.ECycle = 2000L, max.MCycle = 100L, n.decimal = 3L,
n.Quadpts = 31L, Theta.lim = c(-6, 6), Missing = -9,
ParConstraint = FALSE, BiasSE=FALSE)
```

Arguments

data   A `matrix` or `data.frame` consists of dichotomous data (1 for correct and 0 for wrong response), with missing data coded as in Missing (by default, Missing=-9). Each row of data represents a examinne' responses, and each column represents an item.

PriorAlpha   The user specified **normal distribution** prior for the logarithmic weight of the ability in the guessing component (*ln(alpha)*) parameter in the 1PLAG model. Can be:

- A `numeric` with two hyperparameters normal distribution for all *log(alpha)* parameters. By default, PriorAlpha=c(-1.9,1), which means a Normal prior of mean=-1.9 and variance=1 will be used for the logarithmic weight of the ability.
- A `NA`, refers to no priors will be used, so maximum likelihood estimates for the weight of the ability will be obtained.

PriorBeta   The user specified **normal distribution** prior for item difficulty (*beta*) parameters in the 1PLAG and 1PLG model. Can be:

- A `numeric` with two hyperparameters mean and variance of normal distribution for all *beta* parameters. By default, PriorBeta=c(0,4), which means a normal prior of mean=0 and variance=4 will be used for all item difficulty parameters.
- A `NA`, refers to no priors will be used, so maximum likelihood estimates for item difficulty parameter will be obtained.
- A `matrix` with two columns, and each row of matrix consists of two hyperparameters of normal prior (mean and variance) for single item *beta* parameter.

PriorGamma   The user specified **normal distribution** prior for item guessing (*gamma*) parameters in the 1PLAG and 1PLG model. Can be:

- A `numeric` with two hyperparameters mean and variance of normal distribution for all *gamma* parameters. By default, PriorGamma=c(-1.39,0.25), which means a normal prior of mean=-1.39 and variance=0.25 will be used for all item guessing parameters.
- A `NA`, refers to no priors will be used, so maximum likelihood estimates for item guessing parameter will be obtained.
- A `matrix` with two columns, and each row of matrix consists of two hyperparameters of normal prior (mean and variance) for single item *gamma* parameter.

InitialAlpha   The user specified starting value for the weight of the ability in the guessing component (*alpha*) parameters in the 1PLAG model. Can be:

- A `NA` (default), refers to no specified starting values for *alpha* parameter.
- A single number (`numeric`), refers to set this number to be the starting value of *alpha*.

InitialBeta   The user specified starting values for item difficulty (*beta*) parameters in the 1PLAG and 1PLG models. Can be:

- A `NA` (default), refers to no specified starting values for *beta* parameter.
- A single number (`numeric`), refers to set this number to be the starting values of *beta* for all items.
- A `numeric` consists of starting values for each *beta* parameter.

InitialGamma   The user specified starting values for item guessing (*gamma*) parameters in the 1PLAG and 1PLG models. Can be:

- A `NA` (default), refers to no specified starting values for *gamma* parameter.
- A single number (`numeric`), refers to set this number to be the starting values of *gamma* for all items.
- A `numeric` consists of starting values for each *gamma* parameter.

Tol   A single number (`numeric`), refers to convergence threshold for E-step cycles; defaults are 0.0001.

max.ECycle   A single `integer`, refers to maximum number of E-step cycles; defaults are 2000L.

max.MCycle	A single `integer`, refers to maximum number of M-step cycles; defaults are 100L.
n.Quadpts	A single `integer`, refers to number of quadrature points per dimension (must be larger than 5); defaults are 31L.
n.decimal	A single `integer`, refers to number of decimal places when outputs results.
Theta.lim	A `numeric` with two number, refers to the range of integration grid for each dimension; default is c(-6, 6).
Missing	A single number (`numeric`) to indicate which elements are missing; default is -9. The Missing cannot be 0 or 1.
ParConstraint	A logical value to indicate whether estimates parametes in a reasonable range; default is FALSE. If ParConstraint=TRUE: *alpha* in [0, 0.707], *beta* in [-6, 6], *gamma* in [-7, 0].
BiasSE	A logical value to determine whether directly estimating SEs from inversed Hession matrix rather than USEM method, default is FALSE.

Details

One parameter logsitc ability-based guessing (1PLAG) model proposed by San Martín et al.(2006). Let *invlogit(x)=1 / (1 + exp(-x))*:

$$P(x = 1|\theta, \alpha, \beta, \gamma) = invlogit(\theta - \beta) + (1 - invlogit(\theta - \beta)) * invlogit(\alpha * \theta + \gamma)$$

where x=1 is the correct response, *theta* is examinne's ability; *alpha* is the weight of the ability in the guessing component; *beta* and *gamma* are the item difficulty and guessing parameter, respectively. These parameter labels are **capitalized** in program for emphasis.

Value

This function will return a list includes following:

Est.ItemPars A `dataframe` consists of the estimates of *alpha*, *beta* and *gamma* parameters and corresponding estimated standard errors.

Est.Theta A `dataframe` consists of the estimates of *theta* and corresponding estimated standard errors (EAP method).

Loglikelihood The loglikelihood.

Iteration The number of iterations.

EM.Map The parameter estimation history of iterations.

fits.test The model fits information includes G2 test, AIC, BIC and RMSEA.

Elapsed.time The running time of the program.

InitialValues The initial values of item parameters.

References

Guo, S., Wu, T., Zheng, C., & Wang, W.-C. (2018, April). *Bayesian Expectation-Maximization-Maximization for 1PL-AG Model*. Paper presented at the 80th NCME Annual Meeting, New York, NY.

San Martín, E., Del Pino, G., & De Boeck, P. (2006). IRT models for ability-based guessing. *Applied Psychological Measurement*, *30*(3), 183-203. doi:10.1177/0146621605282773

Examples

```
###Example: A brief simulation study###

#generate true values and response matrix
set.seed(10)
library(IRTBEMM)
I=500    #set the number of examinees is 500
J=10       #set the number of items is 10
true.alpha=0.2 #simulate true weight parameters
true.beta=rnorm(J,0,1)     #simulate true difficulty parameters
true.gamma=rnorm(J,-1.39,0.5)    #simulate true guessing parameters
true.th=rnorm(I,0,1)         #simulate true theta parameters
true.par=list(Alpha=true.alpha, Beta=true.beta, Gamma=true.gamma)   #make a list
response=matrix(NA,I,J)        #Create a array to save response data
for (i in 1:I){
  #calucate the probability of 1PLAG
  P=Prob.model(X=true.th[i], Model='1PLAG', Par.est0=true.par)
  response[i,]=rbinom(J,1,P)   #simulate the response
}

#To save example running time, we set the Tol to 0.1
#Obtain the Bayesian modal estimation (BME) using default priors

#Estimate model via BEMM algorithm
bme.res=BEMM.1PLAG(response, Tol=0.1)

bme.res$Est.ItemPars        #show item estimates
bme.res$Est.Theta            #show ability estimates
bme.res$Loglikelihood        #show log-likelihood
bme.res$EM.Map                #show EM iteration history
bme.res$fits.test        #show model fits information


#Obtain the maximum likelihood estimation (MLE) by setting Prior=NA

#Estimate model via EMM algorithm
mle.res=BEMM.1PLAG(response, PriorAlpha=NA, PriorBeta=NA, PriorGamma=NA, Tol=0.1)

mle.res$Est.ItemPars        #show item estimates
mle.res$Est.Theta            #show ability estimates
mle.res$Loglikelihood        #show log-likelihood
mle.res$EM.Map                #show EM iteration history
mle.res$fits.test      #show model fits information
```

BEMM.1PLG *Calibrating 1PLG model via Bayesian Expectation-Maximization-Maximization (BEMM) algorithm.*

Description

This function can estimate the item parameters of the 1PLG model via Bayesian Expectation-Maximization-Maximization (BEMM) algorithm proposed by Guo, Wu, Zheng, & Wang (2018, April). Both Bayesan modal estimates and maximum likelihood estimates are available. In addition, the examinees' ability and a few model fits information can be also obtained through this function.

Usage

```
BEMM.1PLG(data, PriorBeta = c(0, 4), PriorGamma = c(-1.39, 0.25),
InitialBeta = NA, InitialGamma = NA,
Tol = 0.0001, max.ECycle = 2000L, max.MCycle = 100L,
n.decimal = 3L, n.Quadpts = 31L, Theta.lim = c(-6, 6),
Missing = -9, ParConstraint = FALSE, BiasSE=FALSE)
```

Arguments

data   A `matrix` or `data.frame` consists of dichotomous data (1 for correct and 0 for wrong response), with missing data coded as in Missing (by default, Missing=-9). Each row of data represents a examinne' responses, and each column represents an item.

PriorBeta   The user specified **normal distribution** prior for item difficulty (*beta*) parameters in the 1PLAG and 1PLG model. Can be:

  - A `numeric` with two hyperparameters mean and variance of normal distribution for all *beta* parameters. By default, PriorBeta=c(0,4), which means a normal prior of mean=0 and variance=4 will be used for all item difficulty parameters.
  - A `NA`, refers to no priors will be used, so maximum likelihood estimates for item difficulty parameter will be obtained.
  - A `matrix` with two columns, and each row of matrix consists of two hyperparameters of normal prior (mean and variance) for single item *beta* parameter.

PriorGamma   The user specified **normal distribution** prior for item guessing (*gamma*) parameters in the 1PLAG and 1PLG model. Can be:

  - A `numeric` with two hyperparameters mean and variance of normal distribution for all *gamma* parameters. By default, PriorGamma=c(-1.39,0.25), which means a normal prior of mean=-1.39 and variance=0.25 will be used for all item guessing parameters.
  - A `NA`, refers to no priors will be used, so maximum likelihood estimates for item guessing parameter will be obtained.
  - A `matrix` with two columns, and each row of matrix consists of two hyperparameters of normal prior (mean and variance) for single item *gamma* parameter.

InitialBeta   The user specified starting values for item difficulty (*beta*) parameters in the 1PLAG and 1PLG models. Can be:

  - A `NA` (default), refers to no specified starting values for *beta* parameter.

- A single number (numeric), refers to set this number to be the starting values of *beta* for all items.
- A numeric consists of starting values for each *beta* parameter.

InitialGamma   The user specified starting values for item guessing (*gamma*) parameters in the 1PLAG and 1PLG models. Can be:

- A NA (default), refers to no specified starting values for *gamma* parameter.
- A single number (numeric), refers to set this number to be the starting values of *gamma* for all items.
- A numeric consists of starting values for each *gamma* parameter.

Tol   A single number (numeric), refers to convergence threshold for E-step cycles; defaults are 0.0001.

max.ECycle   A single integer, refers to maximum number of E-step cycles; defaults are 2000L.

max.MCycle   A single integer, refers to maximum number of M-step cycles; defaults are 100L.

n.Quadpts   A single integer, refers to number of quadrature points per dimension (must be larger than 5); defaults are 31L.

n.decimal   A single integer, refers to number of decimal places when outputs results.

Theta.lim   A numeric with two number, refers to the range of integration grid for each dimension; default is c(-6, 6).

Missing   A single number (numeric) to indicate which elements are missing; default is -9. The Missing cannot be 0 or 1.

ParConstraint   A logical value to indicate whether estimates parametes in a reasonable range; default is FALSE. If ParConstraint=TRUE: *beta* in [-6, 6], *gamma* in [-7, 0].

BiasSE   A logical value to determine whether directly estimating SEs from inversed Hession matrix rather than USEM method, default is FALSE.

Details

One parameter logsitc guessing (1PLG) model proposed by San Martín et al.(2006). Let *invlogit(x)=1 / (1 + exp(-x))*:

$$P(x = 1|\theta, \beta, \gamma) = invlogit(\theta - \beta) + (1 - invlogit(\theta - \beta)) * invlogit(\gamma)$$

where x=1 is the correct response, *theta* is examinne's ability; *beta* and *gamma* are the item difficulty and guessing parameter, respectively. These parameter labels are **capitalized** in program for emphasis.

Value

This function will return a list includes following:

Est.ItemPars A dataframe consists of the estimates of *beta* and *gamma* parameters and corresponding estimated standard errors.

Est.Theta A dataframe consists of the estimates of *theta* and corresponding estimated standard errors (EAP method).

Loglikelihood The loglikelihood.

Iteration The number of iterations.

EM.Map The parameter estimation history of iterations.

fits.test The model fits information includes G2 test, AIC, BIC and RMSEA.

Elapsed.time The running time of the program.

InitialValues The initial values of item parameters.

References

Guo, S., Wu, T., Zheng, C., & Wang, W.-C. (2018, April). *Bayesian Expectation-Maximization-Maximization for 1PL-AG Model*. Paper presented at the 80th NCME Annual Meeting, New York, NY.

San Martín, E., Del Pino, G., & De Boeck, P. (2006). IRT models for ability-based guessing. *Applied Psychological Measurement*, *30*(3), 183-203. doi:10.1177/0146621605282773

Examples

```
###Example: A brief simulation study###

#generate true values and response matrix
set.seed(10)
library(IRTBEMM)
I=500     #set the number of examinees is 500
J=10        #set the number of items is 10
true.beta=rnorm(J,0,1)     #simulate true difficulty parameters
true.gamma=rnorm(J,-1.39,0.5)     #simulate true guessing parameters
true.th=rnorm(I,0,1)         #simulate true theta parameters
true.par=list(Beta=true.beta, Gamma=true.gamma)   #make a list
response=matrix(NA,I,J)         #Create a array to save response data
for (i in 1:I){
   #caluate the probability of 1PLG
   P=Prob.model(X=true.th[i], Model='1PLG', Par.est0=true.par)
   response[i,]=rbinom(J,1,P)   #simulate the response
}

#To save example running time, we set the Tol to 0.1
#Obtain the Bayesian modal estimation (BME) using default priors

#Estimate model via BEMM algorithm
bme.res=BEMM.1PLG(response, Tol=0.1)

bme.res$Est.ItemPars        #show item estimates
bme.res$Est.Theta           #show ability estimates
bme.res$Loglikelihood       #show log-likelihood
bme.res$EM.Map              #show EM iteration history
bme.res$fits.test       #show model fits information


#Obtain the maximum likelihood estimation (MLE) by setting Prior=NA
```

```
#Estimate model via EMM algorithm
mle.res=BEMM.1PLG(response, PriorBeta=NA, PriorGamma=NA, Tol=0.1)

mle.res$Est.ItemPars        #show item estimates
mle.res$Est.Theta           #show ability estimates
mle.res$Loglikelihood       #show log-likelihood
mle.res$EM.Map              #show EM iteration history
mle.res$fits.test    #show model fits information
```

BEMM.3PL	*Calibrating 3PL model via Bayesian Expectation-Maximization-Maximization (BEMM) algorithm.*

Description

This function can estimate the item parameters of the 3PL model via Bayesian Expectation-Maximization-Maximization (BEMM) algorithm proposed by Guo & Zheng(2019) and Zheng, Meng, Guo, & Liu (2018). Both Bayesan modal estimates and maximum likelihood estimates are available. In addition, the examinees' ability and a few model fits information can be also obtained through this function.

Usage

```
BEMM.3PL(data, PriorA = c(0, 0.25), PriorB = c(0, 4), PriorC = c(4, 16),
InitialA = NA, InitialB = NA, InitialC = NA,
Tol = 0.0001, max.ECycle = 2000L, max.MCycle = 100L,
n.decimal = 3L, n.Quadpts = 31L, Theta.lim = c(-6, 6),
Missing = -9, ParConstraint = FALSE, BiasSE=FALSE)
```

Arguments

data
: A matrix or data.frame consists of dichotomous data (1 for correct and 0 for wrong response), with missing data coded as in Missing (by default, Missing=-9). Each row of data represents a examinne' responses, and each column represents an item.

PriorA
: The user specified **logarithmic normal distribution** prior for item discrimation (*a*) parameters in the 3PL and 4PL models. Can be:

  - A numeric with two hyperparameters mean and variance of logarithmic normal distribution for all *a* parameters. By default, PriorA=c(0,0.25), which means a log normal prior of mean=0 and variance=0.25 will be used for all item discrimation parameters.
  - A NA, refers to no priors will be used, so maximum likelihood estimates for item discrimation parameter will be obtained.
  - A matrix with two columns, and each row of matrix consists of two hyperparameters of log normal prior (mean and variance) for single item *a* parameter.

PriorB   The user specified **normal distribution** prior for item difficulty (*b*) parameters
                in the 3PL and 4PL models. Can be:

- A `numeric` with two hyperparameters mean and variance of normal distribution for all *b* parameters. By default, PriorB=c(0,4), which means a normal prior of mean=0 and variance=4 will be used for all item difficulty parameters.
- A `NA`, refers to no priors will be used, so maximum likelihood estimates for item difficulty parameter will be obtained.
- A `matrix` with two columns, and each row of matrix consists of two hyperparameters of normal prior (mean and variance) for single item *b* parameter.

PriorC   The user specified **Beta(x,y) distribution** prior for item guessing (*c*) parameters
                in the 3PL and 4PL models. Can be:

- A `numeric` with two hyperparameters x and y of Beta distribution for all *c* parameters. By default, PriorC=c(4,16), which means a Beta prior of mean=4/(4+16)=0.2 and variance=0.008 will be used for all item guessing parameters.
- A `NA`, refers to no priors will be used, so maximum likelihood estimates for item guessing parameter will be obtained.
- A `matrix` with two columns, and each row of matrix consists of two hyperparameters of Beta prior (x and y) for single item *c* parameter.

InitialA   The user specified starting values for item discrimation (*a*) parameters in the
                3PL and 4PL models. Can be:

- A `NA` (default), refers to no specified starting values for *a* parameter.
- A single number (`numeric`), refers to set this number to be the starting values of *a* for all items.
- A `numeric` consists of starting values for each *a* parameter.

InitialB   The user specified starting values for item difficulty (*b*) parameters in the 3PL
                and 4PL models. Can be:

- A `NA` (default), refers to no specified starting values for *b* parameter.
- A single number (`numeric`), refers to set this number to be the starting values of *b* for all items.
- A `numeric` consists of starting values for each *b* parameter.

InitialC   The user specified starting values for item guessing (*c*) parameters in the 3PL
                and 4PL models. Can be:

- A `NA` (default), refers to no specified starting values for *c* parameter.
- A single number (`numeric`), refers to set this number to be the starting values of *c* for all items.
- A `numeric` consists of starting values for each *c* parameter.

Tol   A single number (`numeric`), refers to convergence threshold for E-step cycles;
                defaults are 0.0001.

max.ECycle   A single `integer`, refers to maximum number of E-step cycles; defaults are
                2000L.

max.MCycle   A single `integer`, refers to maximum number of M-step cycles; defaults are
                100L.

n.Quadpts	A single `integer`, refers to number of quadrature points per dimension (must be larger than 5); defaults are 31L.
n.decimal	A single `integer`, refers to number of decimal places when outputs results.
Theta.lim	A `numeric` with two number, refers to the range of integration grid for each dimension; default is c(-6, 6).
Missing	A single number (`numeric`) to indicate which elements are missing; default is -9. The Missing cannot be 0 or 1.
ParConstraint	A logical value to indicate whether estimates parametes in a reasonable range; default is FALSE. If ParConstraint=TRUE: *a* in [0.001, 6], *b* in [-6, 6], *c* in [0.0001, 0.5].
BiasSE	A logical value to determine whether directly estimating SEs from inversed Hession matrix rather than USEM method, default is FALSE.

Details

Three parameter logistic (3PL) model proposed by Birnbaum(1968):

$$P(x = 1|\theta, a, b, c) = c + (1 - c)/(1 + exp(-D * a * (\theta - b))),$$

where x=1 is the correct response, *theta* is examinne's ability; *a*, *b* and *c* are the item discrimination, difficulty and guessing parameter, respectively; D is the scaling constant 1.702. These parameter labels are **capitalized** in program for emphasis.

Value

This function will return a list includes following:

Est.ItemPars A `dataframe` consists of the estimates of *a*, *b* and *c* parameters and corresponding estimated standard errors.

Est.Theta A `dataframe` consists of the estimates of *theta* and corresponding estimated standard errors (EAP method).

Loglikelihood The loglikelihood.

Iteration The number of iterations.

EM.Map The parameter estimation history of iterations.

fits.test The model fits information includes G2 test, AIC, BIC and RMSEA.

Elapsed.time The running time of the program.

InitialValues The initial values of item parameters.

References

Birnbaum, A. (1968). *Some latent trait models and their use in inferring an examinee's ability*. In F. M. Lord & M. R. Novick (Eds.), Statistical theories of mental test scores (pp. 395-479). MA: Adison-Wesley.

Guo, S., & Zheng, C. (2019). The Bayesian Expectation-Maximization-Maximization for the 3PLM. *Frontiers in Psychology*, *10*(1175), 1-11. doi:10.3389/fpsyg.2019.01175

Zheng, C., Meng, X., Guo, S., & Liu, Z. (2018). Expectation-Maximization-Maximization: A feasible MLE algorithm for the three-parameter logistic model based on a mixture modeling reformulation. *Frontiers in Psychology*, *8*(2302), 1-10. doi:10.3389/fpsyg.2017.02302

Examples

```
###Example: A brief simulation study###

#generate true values and response matrix
set.seed(10)
library(IRTBEMM)
I=500    #set the number of examinees is 500
J=10     #set the number of items is 10
true.a=runif(J,0.4,2)      #simulate true discrimination parameters
true.b=rnorm(J,0,1)     #simulate true difficulty parameters
true.c=rbeta(J,2,8)       #simulate true guessing parameters
true.th=rnorm(I,0,1)      #simulate true theta parameters
true.par=list(A=true.a, B=true.b, C=true.c)   #make a list
response=matrix(NA,I,J)       #Create a array to save response data
for (i in 1:I){
  #calucate the probability of 3PL
  P=Prob.model(X=true.th[i], Model='3PL', Par.est0=true.par, D=1.702)
  response[i,]=rbinom(J,1,P)   #simulate the response
}

#To save example running time, we set the Tol to 0.1
#Obtain the Bayesian modal estimation (BME) using default priors

#Estimate model via BEMM algorithm
bme.res=BEMM.3PL(response, Tol=0.1)

bme.res$Est.ItemPars      #show item estimates
bme.res$Est.Theta         #show ability estimates
bme.res$Loglikelihood     #show log-likelihood
bme.res$EM.Map            #show EM iteration history
bme.res$fits.test     #show model fits information


#Obtain the maximum likelihood estimation (MLE) by setting Prior=NA

#Estimate model via EMM algorithm
mle.res=BEMM.3PL(response, PriorA=NA, PriorB=NA, PriorC=NA, Tol=0.1)

mle.res$Est.ItemPars      #show item estimates
mle.res$Est.Theta         #show ability estimates
mle.res$Loglikelihood     #show log-likelihood
mle.res$EM.Map            #show EM iteration history
mle.res$fits.test     #show model fits information
```

BEMM.4PL   *Calibrating 4PL model via Bayesian Expectation-Maximization-Maximization (BEMM) algorithm.*

Description

This function can estimate the item parameters of the 4PL model via Bayesian Expectation-Maximization-Maximization (BEMM) algorithm proposed by Zhang, Guo, & Zheng (2018, April). Both Bayesan modal estimates and maximum likelihood estimates are available. In addition, the examinees' ability and a few model fits information can be also obtained through this function.

Usage

```
BEMM.4PL(data, PriorA = c(0, 0.25), PriorB = c(0, 4), PriorC = c(4, 16),
PriorS = c(4, 16), InitialA = NA, InitialB = NA, InitialC = NA,
InitialS = NA, Tol = 0.0001, max.ECycle = 2000L, max.MCycle = 100L,
n.decimal = 3L, n.Quadpts = 31L, Theta.lim = c(-6, 6),
Missing = -9, ParConstraint = FALSE, BiasSE=FALSE)
```

Arguments

data	A `matrix` or `data.frame` consists of dichotomous data (1 for correct and 0 for wrong response), with missing data coded as in Missing (by default, Missing=-9). Each row of data represents a examinne' responses, and each column represents an item.
PriorA	The user specified **logarithmic normal distribution** prior for item discrimation (*a*) parameters in the 3PL and 4PL models. Can be:

- A `numeric` with two hyperparameters mean and variance of logarithmic normal distribution for all *a* parameters. By default, PriorA=c(0,0.25), which means a log normal prior of mean=0 and variance=0.25 will be used for all item discrimation parameters.
- A `NA`, refers to no priors will be used, so maximum likelihood estimates for item discrimation parameter will be obtained.
- A `matrix` with two columns, and each row of matrix consists of two hyperparameters of log normal prior (mean and variance) for single item *a* parameter.

PriorB	The user specified **normal distribution** prior for item difficulty (*b*) parameters in the 3PL and 4PL models. Can be:

- A `numeric` with two hyperparameters mean and variance of normal distribution for all *b* parameters. By default, PriorB=c(0,4), which means a normal prior of mean=0 and variance=4 will be used for all item difficulty parameters.
- A `NA`, refers to no priors will be used, so maximum likelihood estimates for item difficulty parameter will be obtained.
- A `matrix` with two columns, and each row of matrix consists of two hyperparameters of normal prior (mean and variance) for single item *b* parameter.

PriorC	The user specified **Beta(x,y) distribution** prior for item guessing (*c*) parameters in the 3PL and 4PL models. Can be:

- A `numeric` with two hyperparameters x and y of Beta distribution for all *c* parameters. By default, PriorC=c(4,16), which means a Beta prior of mean=4/(4+16)=0.2 and variance=0.008 will be used for all item guessing parameters.

- A `NA`, refers to no priors will be used, so maximum likelihood estimates for item guessing parameter will be obtained.
- A `matrix` with two columns, and each row of matrix consists of two hyper-parameters of Beta prior (x and y) for single item c parameter.

PriorS   The user specified **Beta(x,y) distribution** prior for item slipping (s) parameters in the 4PL model. Can be:

- A `numeric` with two hyperparameters x and y of Beta distribution for all s parameters. By default, PriorS=c(4,16), which means a Beta prior of mean=4/(4+16)=0.2 and variance=0.008 will be used for all item slipping parameters.
- A `NA`, refers to no priors will be used, so maximum likelihood estimates for item slipping parameter will be obtained.
- A `matrix` with two columns, and each row of matrix consists of two hyper-parameters of Beta prior (x and y) for single item s parameter.

InitialA   The user specified starting values for item discrimation (a) parameters in the 3PL and 4PL models. Can be:

- A `NA` (default), refers to no specified starting values for a parameter.
- A single number (`numeric`), refers to set this number to be the starting values of a for all items.
- A `numeric` consists of starting values for each a parameter.

InitialB   The user specified starting values for item difficulty (b) parameters in the 3PL and 4PL models. Can be:

- A `NA` (default), refers to no specified starting values for b parameter.
- A single number (`numeric`), refers to set this number to be the starting values of b for all items.
- A `numeric` consists of starting values for each b parameter.

InitialC   The user specified starting values for item guessing (c) parameters in the 3PL and 4PL models. Can be:

- A `NA` (default), refers to no specified starting values for c parameter.
- A single number (`numeric`), refers to set this number to be the starting values of c for all items.
- A `numeric` consists of starting values for each c parameter.

InitialS   The user specified starting values for item slipping (s) parameters in the 4PL model. Can be:

- A `NA` (default), refers to no specified starting values for s parameter.
- A single number (`numeric`), refers to set this number to be the starting values of s for all items.
- A `numeric` consists of starting values for each s parameter.

Tol   A single number (`numeric`), refers to convergence threshold for E-step cycles; defaults are 0.0001.

max.ECycle   A single `integer`, refers to maximum number of E-step cycles; defaults are 2000L.

max.MCycle	A single `integer`, refers to maximum number of M-step cycles; defaults are 100L.
n.Quadpts	A single `integer`, refers to number of quadrature points per dimension (must be larger than 5); defaults are 31L.
n.decimal	A single `integer`, refers to number of decimal places when outputs results.
Theta.lim	A `numeric` with two number, refers to the range of integration grid for each dimension; default is c(-6, 6).
Missing	A single number (`numeric`) to indicate which elements are missing; default is -9. The Missing cannot be 0 or 1.
ParConstraint	A logical value to indicate whether estimates parametes in a reasonable range; default is FALSE. If ParConstraint=TRUE: *a* in [0.001, 6], *b* in [-6, 6], *c* in [0.0001, 0.5], *s* in [0.0001, 0.5].
BiasSE	A logical value to determine whether directly estimating SEs from inversed Hession matrix rather than USEM method, default is FALSE.

Details

Four parameter logistic (4PL) model proposed by Barton & Lord's (1981). Transfer the unslipping (upper asymptote) parameter *d* to slipping parameter *s* by set *s=1-d*:

$$P(x = 1|\theta, a, b, c, s) = c + (1 - s - c)/(1 + exp(-D * a * (\theta - b))),$$

where x=1 is the correct response; *theta* is examinne's ability. *a*, *b*, *c* and *s* are the item discrimination, difficulty guessing and slipping parameter, respectively; D is the scaling constant 1.702. These parameter labels are **capitalized** in program for emphasis.

Value

This function will return a list includes following:

Est.ItemPars A `dataframe` consists of the estimates of *a*, *b*, *c* and *s* parameters and corresponding estimated standard errors.

Est.Theta A `dataframe` consists of the estimates of *theta* and corresponding estimated standard errors (EAP method).

Loglikelihood The loglikelihood.

Iteration The number of iterations.

EM.Map The parameter estimation history of iterations.

fits.test The model fits information includes G2 test, AIC, BIC and RMSEA.

Elapsed.time The running time of the program.

InitialValues The initial values of item parameters.

References

Barton, M. A., & Lord, F. M. (1981). An upper asymptote for the three-parameter logistic item response model. *ETS Research Report Series*, *1981*(1), 1-8. doi:10.1002/j.23338504.1981.tb01255.x

Zhang, C., Guo, S., & Zheng, C. (2018, April). *Bayesian Expectation-Maximization-Maximization Algorithm for the 4PLM*. Paper presented at the 80th NCME Annual Meeting, New York, NY.

Examples

```
###Example: A brief simulation study###

#generate true values and response matrix
set.seed(10)
library(IRTBEMM)
I=500    #set the number of examinees is 500
J=10     #set the number of items is 10
true.a=runif(J,0.4,2)     #simulate true discrimination parameters
true.b=rnorm(J,0,1)       #simulate true difficulty parameters
true.c=rbeta(J,2,8)       #simulate true guessing parameters
true.s=rbeta(J,2,8)       #simulate true slipping parameters
true.th=rnorm(I,0,1)      #simulate true theta parameters
true.par=list(A=true.a, B=true.b, C=true.c, S=true.s)   #make a list
response=matrix(NA,I,J)        #Create a array to save response data
for (i in 1:I){
  #calucate the probability of 4PL
  P=Prob.model(X=true.th[i], Model='4PL', Par.est0=true.par, D=1.702)
  response[i,]=rbinom(J,1,P)   #simulate the response
}

#To save example running time, we set the Tol to 0.1
#Obtain the Bayesian modal estimation (BME) using default priors

#Estimate model via BEMM algorithm
bme.res=BEMM.4PL(response, Tol=0.1)

bme.res$Est.ItemPars      #show item estimates
bme.res$Est.Theta         #show ability estimates
bme.res$Loglikelihood     #show log-likelihood
bme.res$EM.Map            #show EM iteration history
bme.res$fits.test     #show model fits information


#Obtain the maximum likelihood estimation (MLE) by setting Prior=NA

#Estimate model via EMM algorithm
mle.res=BEMM.4PL(response, Tol=0.1,
PriorA=NA, PriorB=NA, PriorC=NA, PriorS=NA)

mle.res$Est.ItemPars      #show item estimates
mle.res$Est.Theta         #show ability estimates
mle.res$Loglikelihood     #show log-likelihood
mle.res$EM.Map            #show EM iteration history
mle.res$fits.test     #show model fits information
```

Input.Checking   *Checking user speciflied input variables.*

Description

Based on the given model, checking whether user specified input variables are correct. If the input variables are acceptable, this function will format them and then return them as a `list`. Otherwise, this function will return a error message to indicate which variables are unacceptable.

Usage

```
Input.Checking(Model, data,
PriorA=c(0,0.25), PriorB=c(0,4), PriorC=c(4,16), PriorS=c(4,16),
PriorAlpha=c(-1.9,1), PriorBeta=c(0,4), PriorGamma=c(-1.39,0.25),
InitialA=NA, InitialB=NA, InitialC=NA, InitialS=NA,
InitialAlpha=NA, InitialBeta=NA, InitialGamma=NA,
Tol=0.0001, max.ECycle=1000L, max.MCycle=100L, n.Quadpts=31L, n.decimal=3L,
Theta.lim=c(-6,6), Missing=-9, ParConstraint=FALSE, BiasSE=FALSE)
```

Arguments

Model
A `character` to declare the type of items to be modeled. The parameter labels follow conventional use, can be:

- '3PL' - Three parameter logistic (3PL) model proposed by Birnbaum(1968):

$$P(x = 1|\theta, a, b, c) = c + (1 - c)/(1 + exp(-D * a * (\theta - b)))$$

  where x=1 is the correct response, *theta* is examinne's ability; *a*, *b* and *c* are the item discrimination, difficulty and guessing parameter, respectively; D is the scaling constant 1.702.

- '4PL' - Four parameter logistic (4PL) model proposed by Barton & Lord's(1981). Transfer the unslipping (upper asymptote) parameter *d* to slipping parameter *s* by set *s=1-d*:

$$P(x = 1|\theta, a, b, c, s) = c + (1 - s - c)/(1 + exp(-D * a * (\theta - b)))$$

  where x=1 is the correct response; *theta* is examinne's ability. *a*, *b*, *c* and *s* are the item discrimination, difficulty guessing and slipping parameter, respectively; D is the scaling constant 1.702.

- '1PLG' - One parameter logsitc guessing (1PLG) model proposed by San Martín et al.(2006). Let *invlogit(x)=1 / (1 + exp(-x))*:

$$P(x = 1|\theta, \beta, \gamma) = invlogit(\theta - \beta) + (1 - invlogit(\theta - \beta)) * invlogit(\gamma)$$

  where x=1 is the correct response, *theta* is examinne's ability; *beta* and *gamma* are the item difficulty and guessing parameter, respectively.

- '1PLAG' - One parameter logsitc ability-based guessing (1PLAG) model proposed by San Martín et al.(2006). Let *invlogit(x)=1 / (1 + exp(-x))*:

$$P(x = 1|\theta, \alpha, \beta, \gamma) = invlogit(\theta - \beta) + (1 - invlogit(\theta - \beta)) * invlogit(\alpha * \theta + \gamma)$$

  where x=1 is the correct response, *theta* is examinne's ability; *alpha* is the weight of the ability in the guessing component; *beta* and *gamma* are the item difficulty and guessing parameter, respectively.

These parameter labels are **capitalized** in program for emphasis.

data   A matrix or data.frame consists of dichotomous data (1 for correct and 0 for wrong response), with missing data coded as in Missing (by default, Missing=-9). Each row of data represents a examinne' responses, and each column represents an item.

PriorA   The user specified **logarithmic normal distribution** prior for item discrimation (*a*) parameters in the 3PL and 4PL models. Can be:

- A numeric with two hyperparameters mean and variance of logarithmic normal distribution for all *a* parameters. By default, PriorA=c(0,0.25), which means a log normal prior of mean=0 and variance=0.25 will be used for all item discrimation parameters.
- A NA, refers to no priors will be used, so maximum likelihood estimates for item discrimation parameter will be obtained.
- A matrix with two columns, and each row of matrix consists of two hyperparameters of log normal prior (mean and variance) for single item *a* parameter.

PriorB   The user specified **normal distribution** prior for item difficulty (*b*) parameters in the 3PL and 4PL models. Can be:

- A numeric with two hyperparameters mean and variance of normal distribution for all *b* parameters. By default, PriorB=c(0,4), which means a normal prior of mean=0 and variance=4 will be used for all item difficulty parameters.
- A NA, refers to no priors will be used, so maximum likelihood estimates for item difficulty parameter will be obtained.
- A matrix with two columns, and each row of matrix consists of two hyperparameters of normal prior (mean and variance) for single item *b* parameter.

PriorC   The user specified **Beta(x,y) distribution** prior for item guessing (*c*) parameters in the 3PL and 4PL models. Can be:

- A numeric with two hyperparameters x and y of Beta distribution for all *c* parameters. By default, PriorC=c(4,16), which means a Beta prior of mean=4/(4+16)=0.2 and variance=0.008 will be used for all item guessing parameters.
- A NA, refers to no priors will be used, so maximum likelihood estimates for item guessing parameter will be obtained.
- A matrix with two columns, and each row of matrix consists of two hyperparameters of Beta prior (x and y) for single item *c* parameter.

PriorS   The user specified **Beta(x,y) distribution** prior for item slipping (*s*) parameters in the 4PL model. Can be:

- A numeric with two hyperparameters x and y of Beta distribution for all *s* parameters. By default, PriorS=c(4,16), which means a Beta prior of mean=4/(4+16)=0.2 and variance=0.008 will be used for all item slipping parameters.
- A NA, refers to no priors will be used, so maximum likelihood estimates for item slipping parameter will be obtained.

- A matrix with two columns, and each row of matrix consists of two hyper-parameters of Beta prior (x and y) for single item *s* parameter.

PriorAlpha The user specified **normal distribution** prior for the logarithmic weight of the ability in the guessing component (*ln(alpha)*) parameter in the 1PLAG model. Can be:

- A numeric with two hyperparameters normal distribution for all *log(alpha)* parameters. By default, PriorAlpha=c(-1.9,1), which means a Normal prior of mean=-1.9 and variance=1 will be used for the logarithmic weight of the ability.
- A NA, refers to no priors will be used, so maximum likelihood estimates for the weight of the ability will be obtained.

PriorBeta The user specified **normal distribution** prior for item difficulty (*beta*) parameters in the 1PLAG and 1PLG model. Can be:

- A numeric with two hyperparameters mean and variance of normal distribution for all *beta* parameters. By default, PriorBeta=c(0,4), which means a normal prior of mean=0 and variance=4 will be used for all item difficulty parameters.
- A NA, refers to no priors will be used, so maximum likelihood estimates for item difficulty parameter will be obtained.
- A matrix with two columns, and each row of matrix consists of two hyperparameters of normal prior (mean and variance) for single item *beta* parameter.

PriorGamma The user specified **normal distribution** prior for item guessing (*gamma*) parameters in the 1PLAG and 1PLG model. Can be:

- A numeric with two hyperparameters mean and variance of normal distribution for all *gamma* parameters. By default, PriorGamma=c(-1.39,0.25), which means a normal prior of mean=-1.39 and variance=0.25 will be used for all item guessing parameters.
- A NA, refers to no priors will be used, so maximum likelihood estimates for item guessing parameter will be obtained.
- A matrix with two columns, and each row of matrix consists of two hyperparameters of normal prior (mean and variance) for single item *gamma* parameter.

InitialA The user specified starting values for item discrimation (*a*) parameters in the 3PL and 4PL models. Can be:

- A NA (default), refers to no specified starting values for *a* parameter.
- A single number (numeric), refers to set this number to be the starting values of *a* for all items.
- A numeric consists of starting values for each *a* parameter.

InitialB The user specified starting values for item difficulty (*b*) parameters in the 3PL and 4PL models. Can be:

- A NA (default), refers to no specified starting values for *b* parameter.
- A single number (numeric), refers to set this number to be the starting values of *b* for all items.

- A numeric consists of starting values for each *b* parameter.

InitialC   The user specified starting values for item guessing (*c*) parameters in the 3PL and 4PL models. Can be:

- A NA (default), refers to no specified starting values for *c* parameter.
- A single number (numeric), refers to set this number to be the starting values of *c* for all items.
- A numeric consists of starting values for each *c* parameter.

InitialS   The user specified starting values for item slipping (*s*) parameters in the 4PL model. Can be:

- A NA (default), refers to no specified starting values for *s* parameter.
- A single number (numeric), refers to set this number to be the starting values of *s* for all items.
- A numeric consists of starting values for each *s* parameter.

InitialAlpha   The user specified starting value for the weight of the ability in the guessing component (*alpha*) parameters in the 1PLAG model. Can be:

- A NA (default), refers to no specified starting values for *alpha* parameter.
- A single number (numeric), refers to set this number to be the starting value of *alpha*.

InitialBeta   The user specified starting values for item difficulty (*beta*) parameters in the 1PLAG and 1PLG models. Can be:

- A NA (default), refers to no specified starting values for *beta* parameter.
- A single number (numeric), refers to set this number to be the starting values of *beta* for all items.
- A numeric consists of starting values for each *beta* parameter.

InitialGamma   The user specified starting values for item guessing (*gamma*) parameters in the 1PLAG and 1PLG models. Can be:

- A NA (default), refers to no specified starting values for *gamma* parameter.
- A single number (numeric), refers to set this number to be the starting values of *gamma* for all items.
- A numeric consists of starting values for each *gamma* parameter.

Tol   A single number (numeric), refers to convergence threshold for E-step cycles; defaults are 0.0001.

max.ECycle   A single integer, refers to maximum number of E-step cycles; defaults are 2000L.

max.MCycle   A single integer, refers to maximum number of M-step cycles; defaults are 100L.

n.Quadpts   A single integer, refers to number of quadrature points per dimension (must be larger than 5); defaults are 31L.

n.decimal   A single integer, refers to number of decimal places when outputs results.

Theta.lim   A numeric with two number, refers to the range of integration grid for each dimension; default is c(-6, 6).

Missing	A single number (`numeric`) to indicate which elements are missing; default is -9. The Missing cannot be 0 or 1.
ParConstraint	A logical value to indicate whether estimates parametes in a reasonable range; default is FALSE. If ParConstraint=TRUE: *a* in [0.001, 6], *b* in [-6, 6], *c* in [0.0001, 0.5], *s* in [0.0001, c], *alpha* in [0, 0.707], *beta* in [-6, 6], *gamma* in [-7, 0].
BiasSE	A logical value to determine whether directly estimating SEs from inversed Hession matrix rather than USEM method, default is FALSE.

References

Barton, M. A., & Lord, F. M. (1981). An upper asymptote for the three-parameter logistic item response model. *ETS Research Report Series*, *1981*(1), 1-8. doi:10.1002/j.23338504.1981.tb01255.x

Birnbaum, A. (1968). *Some latent trait models and their use in inferring an examinee's ability*. In F. M. Lord & M. R. Novick (Eds.), Statistical theories of mental test scores (pp. 395-479). MA: Adison-Wesley.

San Martín, E., Del Pino, G., & De Boeck, P. (2006). IRT models for ability-based guessing. *Applied Psychological Measurement*, *30*(3), 183-203. doi:10.1177/0146621605282773

Examples

```
#An example to show the error message when the variance of a normal prior is negative.
library(IRTBEMM)
#generate a response matrix with 1000 examinees and 10 items randomly
data=matrix(rbinom(10000,1,0.5), 1000, 10)
#test whether variable data is correct.
res=Input.Checking('3PL',data)
```

Prob.model	*Calculate the probabilites based on a given model and parameters.*

Description

Based on the given model, return the correct probabilities of a single examinne with ability X answering each item.

Usage

```
Prob.model(X, Model, Par.est0, D=1.702)
```

Arguments

X	A `numeric` with length=1 consists of an examinee's ability theta.
Model	A `character` to declare the type of items to be modeled. The parameter labels follow conventional uses, can be:

- '3PL' - Three parameter logistic (3PL) model proposed by Birnbaum(1968):

$$P(x = 1|\theta, a, b, c) = c + (1 - c)/(1 + exp(-D * a * (\theta - b)))$$

where x=1 is the correct response, *theta* is examinne's ability; *a*, *b* and *c* are the item discrimination, difficulty and guessing parameter, respectively; D is the scaling constant 1.702.

- '4PL' - Four parameter logistic (4PL) model proposed by Barton & Lord's(1981). Transfer the unslipping (upper asymptote) parameter *d* to slipping parameter *s* by set *s=1-d*:

$$P(x = 1|\theta, a, b, c, s) = c + (1 - s - c)/(1 + exp(-D * a * (\theta - b)))$$

where x=1 is the correct response; *theta* is examinne's ability. *a*, *b*, *c* and *s* are the item discrimination, difficulty guessing and slipping parameter, respectively; D is the scaling constant 1.702.

- '1PLG' - One parameter logsitc guessing (1PLG) model proposed by San Martín et al.(2006). Let *invlogit(x)=1 / (1 + exp(-x))*:

$$P(x = 1|\theta, \beta, \gamma) = invlogit(\theta - \beta) + (1 - invlogit(\theta - \beta)) * invlogit(\gamma)$$

where x=1 is the correct response, *theta* is examinne's ability; *beta* and *gamma* are the item difficulty and guessing parameter, respectively.

- '1PLAG' - One parameter logsitc ability-based guessing (1PLAG) model proposed by San Martín et al.(2006). Let *invlogit(x)=1 / (1 + exp(-x))*:

$$P(x = 1|\theta, \alpha, \beta, \gamma) = invlogit(\theta - \beta) + (1 - invlogit(\theta - \beta)) * invlogit(\alpha * \theta + \gamma)$$

where x=1 is the correct response, *theta* is examinne's ability; *alpha* is the weight of the ability in the guessing component; *beta* and *gamma* are the item difficulty and guessing parameter, respectively.

These parameter labels are **capitalized** in program for emphasis.

Par.est0   A list that consists of item parameters for each item based on the given model. Can be:

- For 3PL model, **list(A, B, C)** - A, B, C are numeric refer to item discrimination, difficulty and pseudo guessing parameters for each item, respectively.
- For 4PL model, **list(A, B, C, S)** - A, B, C, S are numeric refer to item discrimination, difficulty, pseudo guessing and slipping parameters for each item, respectively.
- For 1PLG model, **list(Beta, Gamma)** - Beta, Gamma are numeric refer to item difficulty and guessing (on the logistic scales) parameters for each item, respectively.
- For 1PLAG model, **list(Alpha, Beta, Gamma)** - Alpha refers to the weight of the ability in the guessing component, and Beta and Gamma are numeric refer to item difficulty and guessing (on the logistic scales) parameters for each item, respectively.

Please note these **capitalized** parameter lables are transformed from the Model section.

D   A single numeric refers to the scaling constant only used in the 3PL and 4PL model. By default, D=1.702.

Value

A `numeric` consists of the correct probabilities of a single examinne with ability X answering each item.

References

Barton, M. A., & Lord, F. M. (1981). An upper asymptote for the three-parameter logistic item response model. *ETS Research Report Series*, *1981*(1), 1-8. doi:10.1002/j.23338504.1981.tb01255.x

Birnbaum, A. (1968). *Some latent trait models and their use in inferring an examinee's ability*. In F. M. Lord & M. R. Novick (Eds.), Statistical theories of mental test scores (pp. 395-479). MA: Adison-Wesley.

San Martín, E., Del Pino, G., & De Boeck, P. (2006). IRT models for ability-based guessing. *Applied Psychological Measurement*, *30*(3), 183-203. doi:10.1177/0146621605282773

Examples

```
#Obtain the correct probabilities of five 3PL model items when theta=1.2 and D=1.702.
library(IRTBEMM)
th=1.2                          #Examinee's ability parameter theta
A=c(1.5, 2, 0.5, 1.2, 0.4)      #item discrimination parameters
B=c(-0.5, 0, 1.5, 0.3, 2.8)     #item difficulty parameters
C=c(0.1, 0.2, 0.3, 0.15, 0.25)  #item pseudo guessing parameters
Par3PL=list(A=A, B=B, C=C)      #Create a list for 3PL
P.3pl=Prob.model(X=th, Model='3PL', Par.est0=Par3PL)   #Obtain the 3PL probabilities


#Obtain the correct probabilities of five 4PL model items when theta=1.2 and D=1.
S=c(0.3, 0.1, 0.13, 0.09, 0.05)  #item pseudo slipping parameters
Par4PL=list(A=A, B=B, C=C, S=S)  #Create a list for 4PL
P.4pl=Prob.model(X=th, Model='4PL', Par.est0=Par4PL, D=1)   #Obtain the 4PL probabilities


#Obtain the correct probabilities of three 1PLG model items when theta=0.3.
th=0.3
Beta=c(0.8, -1.9, 2.4)
Gamma=c(-1.31, -0.89, -0.18)
Par1PLG=list(Beta=Beta, Gamma=Gamma)              #Create a list for 1PLG
P.1plg=Prob.model(X=th, Model='1PLG', Par.est0=Par1PLG)   #Obtain the 1PLG probabilities


#Obtain the correct probabilities of three 1PLAG model items when theta=0.3.
Alpha=0.2
Par1PLAG=list(Alpha=Alpha, Beta=Beta, Gamma=Gamma)      #Create a list for 1PLAG
P.1plag=Prob.model(X=th, Model='1PLAG', Par.est0=Par1PLAG) #Obtain the 1PLAG probabilities
```

Index