Package 'FMP'

February 5, 2016

Version 1.3

Date 2016-01-28

Title Filtered Monotonic Polynomial IRT Models

Author Niels G. Waller <nwaller@umn.edu>

Maintainer Niels G. Waller <nwaller@umn.edu>

Depends R (>= 3.0)

Description Estimates Filtered Monotonic Polynomial IRT Models as described by Liang and Browne (2015) <DOI:10.3102/1076998614556816>.

License GPL (>=2)

R topics documented:

`eap` *Compute eap trait estimates for FMP and FUP models*

Description

Compute eap trait estimates for items fit by filtered monotonic polynomial IRT models.

Usage

```
eap(data, bParams, NQuad = 21, priorVar = 2, mintheta = -4, maxtheta = 4)
```

Arguments

`data` N(subjects)-by-p(items) matrix of 0/1 item response data.

`bParams` A p-by-9 matrix of FMP or FUP item parameters and model designations. Columns 1 - 8 hold the (possibly zero valued) polynomial coefficients; column 9 holds the value of `k`.

`NQuad` Number of quadrature points used to calculate the eap estimates.

`priorVar` Variance of the normal prior for the eap estimates. The prior mean equals 0.

`mintheta, maxtheta` NQuad quadrature points will be evenly spaced between `mintheta` and `maxtheta`

Value

`eap trait estimates.`

Author(s)

Niels Waller

Examples

```
## this example demonstrates how to calculate
## eap trait estimates for a scale composed of items
## that have been fit to FMP models of different
## degree

NSubjects <- 2000

## Assume that
## items 1 - 5 fit a k=0 model,
## items 6 - 10 fit a k=1 model, and
## items 11 - 15 fit a k=2 model.


 itmParameters <- matrix(c(
  #  b0    b1     b2    b3    b4  b5, b6, b7,  k
  -1.05, 1.63,  0.00, 0.00, 0.00,  0,     0,  0,   0, #1
  -1.97, 1.75,  0.00, 0.00, 0.00,  0,     0,  0,   0, #2
  -1.77, 1.82,  0.00, 0.00, 0.00,  0,     0,  0,   0, #3
  -4.76, 2.67,  0.00, 0.00, 0.00,  0,     0,  0,   0, #4
  -2.15, 1.93,  0.00, 0.00, 0.00,  0,     0,  0,   0, #5
  -1.25, 1.17, -0.25, 0.12, 0.00,  0,     0,  0,   1, #6
   1.65, 0.01,  0.02, 0.03, 0.00,  0,     0,  0,   1, #7
  -2.99, 1.64,  0.17, 0.03, 0.00,  0,     0,  0,   1, #8
  -3.22, 2.40, -0.12, 0.10, 0.00,  0,     0,  0,   1, #9
  -0.75, 1.09, -0.39, 0.31, 0.00,  0,     0,  0,   1, #10
  -1.21, 9.07,  1.20,-0.01,-0.01,  0.01,  0,  0,   2, #11
  -1.92, 1.55, -0.17, 0.50,-0.01,  0.01,  0,  0,   2, #12
  -1.76, 1.29, -0.13, 1.60,-0.01,  0.01,  0,  0,   2, #13
  -2.32, 1.40,  0.55, 0.05,-0.01,  0.01,  0,  0,   2, #14
  -1.24, 2.48, -0.65, 0.60,-0.01,  0.01,  0,  0,   2),#15
  15, 9, byrow=TRUE)

# generate data using the above item parameters
```

```
ex1.data<-genFMPData(NSubj = NSubjects, bParams = itmParameters,
                    seed = 345)$data


## calculate eap estimates for mixed models
thetaEAP<-eap(data = ex1.data, bParams = itmParameters,
                   NQuad = 25, priorVar = 2,
                   mintheta = -4, maxtheta = 4)

## compare eap estimates with initial theta surrogates

if(FALSE){     #set to TRUE to see plot

  thetaInit <- svdNorm(ex1.data)
  plot(thetaInit,thetaEAP, xlim = c(-3.5,3.5),
                         ylim = c(-3.5,3.5),
                         xlab = "Initial theta surrogates",
                         ylab = "EAP trait estimates (Mixed models)")
}
```

---

erf — *Utility fnc to compute the components for an empirical response function*

Description

Utility function to compute empirical response functions.

Usage

```
erf(theta, data, whichItem, min = -3, max = 3, Ncuts = 12)
```

Arguments

theta	Vector of estimated latent trait scores.
data	A matrix of binary item responses.
whichItem	Data for an erf will be generated for whichItem.
min	Default = -3. Minimum value of theta.
max	Default = 3. Maximum value of theta.
Ncuts	Number of score groups for erf.

Value

probs	A vector (of length Ncuts) of bin response probabilities for the empirical response function.
centers	A vector of bin centers.
Ni	Bin sample sizes.
se.p	Standard errors of the estimated bin response probabilities.

Author(s)

Niels Waller

Examples

```
NSubj <- 2000

#generate sample k=1 FMP  data
b <- matrix(c(
    #b0    b1     b2    b3      b4   b5  b6  b7  k
  1.675, 1.974, -0.068, 0.053,  0,  0,  0,  0, 1,
  1.550, 1.805, -0.230, 0.032,  0,  0,  0,  0, 1,
  1.282, 1.063, -0.103, 0.003,  0,  0,  0,  0, 1,
  0.704, 1.376, -0.107, 0.040,  0,  0,  0,  0, 1,
  1.417, 1.413,  0.021, 0.000,  0,  0,  0,  0, 1,
 -0.008, 1.349, -0.195, 0.144,  0,  0,  0,  0, 1,
  0.512, 1.538, -0.089, 0.082,  0,  0,  0,  0, 1,
  0.122, 0.601, -0.082, 0.119,  0,  0,  0,  0, 1,
  1.801, 1.211,  0.015, 0.000,  0,  0,  0,  0, 1,
 -0.207, 1.191,  0.066, 0.033,  0,  0,  0,  0, 1,
 -0.215, 1.291, -0.087, 0.029,  0,  0,  0,  0, 1,
  0.259, 0.875,  0.177, 0.072,  0,  0,  0,  0, 1,
 -0.423, 0.942,  0.064, 0.094,  0,  0,  0,  0, 1,
  0.113, 0.795,  0.124, 0.110,  0,  0,  0,  0, 1,
  1.030, 1.525,  0.200, 0.076,  0,  0,  0,  0, 1,
  0.140, 1.209,  0.082, 0.148,  0,  0,  0,  0, 1,
  0.429, 1.480, -0.008, 0.061,  0,  0,  0,  0, 1,
  0.089, 0.785, -0.065, 0.018,  0,  0,  0,  0, 1,
 -0.516, 1.013,  0.016, 0.023,  0,  0,  0,  0, 1,
  0.143, 1.315, -0.011, 0.136,  0,  0,  0,  0, 1,
  0.347, 0.733, -0.121, 0.041,  0,  0,  0,  0, 1,
 -0.074, 0.869,  0.013, 0.026,  0,  0,  0,  0, 1,
  0.630, 1.484, -0.001, 0.000,  0,  0,  0,  0, 1),
  nrow=23, ncol=9, byrow=TRUE)

theta <- rnorm(NSubj)
data<-genFMPData(NSubj = NSubj, bParam = b, theta = theta, seed = 345)$data

erfItem1 <- erf(theta, data, whichItem = 1, min = -3, max = 3, Ncuts = 12)

plot( erfItem1$centers, erfItem1$probs, type="b",
      main="Empirical Response Function",
      xlab = expression(theta),
      ylab="Probability",
      cex.lab=1.5)
```

FMP *Estimate the coefficients of a filtered monotonic polynomial IRT model*

Description

Estimate the coefficients of a filtered monotonic polynomial IRT model.

Usage

```
FMP(data, thetaInit, item, startvals, k, eps = 1e-06)
```

Arguments

`data`	N(subjects)-by-p(items) matrix of 0/1 item response data.
`thetaInit`	Initial theta surrogates (e.g., calculated by svdNorm).
`item`	Item number for coefficient estimation.
`startvals`	Start values for function minimization. Start values are in the gamma metric (see Liang & Browne, 2015)
`k`	Order of monotonic polynomial = 2k+1 (see Liang & Browne, 2015). k can equal 0, 1, 2, or 3.
`eps`	Step size for gradient approximation, default = 1e-6. If a convergence failure occurs during function optimization reducing the value of eps will often produce a converged solution.

Details

As described by Liang and Browne (2015), the filtered polynomial model (FMP) is a quasi-parametric IRT model in which the IRF is a composition of a logistic function and a polynomial function, $m(\theta)$, of degree 2k + 1. When k = 0, $m(\theta) = b0 + b1\theta$ (the slope intercept form of the 2PL). When k = 1, 2k + 1 equals 3 resulting in $m(\theta) = b0 + b_1\theta + b_2\theta^2 + b_3\theta^3$. Acceptable values of k = 0,1,2,3. According to Liang and Browne, the "FMP IRF may be used to approximate any IRF with a continuous derivative arbitrarily closely by increasing the number of parameters in the monotonic polynomial" (2015, p. 2) The FMP model assumes that the IRF is monotonically increasing, bounded by 0 and 1, and everywhere differentiable with respect to theta (the latent trait).

Value

`b`	Vector of polynomial coefficients.
`gamma`	Polynomial coefficients in gamma metric (see Liang & Browne, 2015).
`FHAT`	Function value at convergence.
`counts`	Number of function evaluations during minimization (see optim documentation for further details).
`AIC`	Pseudo scaled Akaike Information Criterion (AIC). Candidate models that produce the smallest AIC suggest the optimal number of parameters given the sample size. Scaling is accomplished by dividing the non-scaled AIC by sample size.
`BIC`	Pseudo scaled Bayesian Information Criterion (BIC). Candidate models that produce the smallest BIC suggest the optimal number of parameters given the sample size. Scaling is accomplished by dividing the non-scaled BIC by sample size.
`convergence`	Convergence = 0 indicates that the optimization algorithm converged; convergence=1 indicates that the optimization failed to converge.

Author(s)

Niels Waller

References

Liang, L. & Browne, M. W. (2015). A quasi-parametric method for fitting flexible item response functions. *Journal of Educational and Behavioral Statistics, 40*, 5–34.

Examples

```
## Not run:
## In this example we will generate 2000 item response vectors
## for a k = 1 order filtered polynomial model and then recover
## the estimated item parameters with the FMP function.

k <- 1  # order of polynomial

NSubjects <- 2000


## generate a sample of 2000 item response vectors
## for a k = 1 FMP model using the following
## coefficients
b <- matrix(c(
   #b0     b1      b2     b3   b4  b5  b6  b7  k
  1.675, 1.974, -0.068, 0.053,  0,  0,  0,  0, 1,
  1.550, 1.805, -0.230, 0.032,  0,  0,  0,  0, 1,
  1.282, 1.063, -0.103, 0.003,  0,  0,  0,  0, 1,
  0.704, 1.376, -0.107, 0.040,  0,  0,  0,  0, 1,
  1.417, 1.413,  0.021, 0.000,  0,  0,  0,  0, 1,
 -0.008, 1.349, -0.195, 0.144,  0,  0,  0,  0, 1,
  0.512, 1.538, -0.089, 0.082,  0,  0,  0,  0, 1,
  0.122, 0.601, -0.082, 0.119,  0,  0,  0,  0, 1,
  1.801, 1.211,  0.015, 0.000,  0,  0,  0,  0, 1,
 -0.207, 1.191,  0.066, 0.033,  0,  0,  0,  0, 1,
 -0.215, 1.291, -0.087, 0.029,  0,  0,  0,  0, 1,
  0.259, 0.875,  0.177, 0.072,  0,  0,  0,  0, 1,
 -0.423, 0.942,  0.064, 0.094,  0,  0,  0,  0, 1,
  0.113, 0.795,  0.124, 0.110,  0,  0,  0,  0, 1,
  1.030, 1.525,  0.200, 0.076,  0,  0,  0,  0, 1,
  0.140, 1.209,  0.082, 0.148,  0,  0,  0,  0, 1,
  0.429, 1.480, -0.008, 0.061,  0,  0,  0,  0, 1,
  0.089, 0.785, -0.065, 0.018,  0,  0,  0,  0, 1,
 -0.516, 1.013,  0.016, 0.023,  0,  0,  0,  0, 1,
  0.143, 1.315, -0.011, 0.136,  0,  0,  0,  0, 1,
  0.347, 0.733, -0.121, 0.041,  0,  0,  0,  0, 1,
 -0.074, 0.869,  0.013, 0.026,  0,  0,  0,  0, 1,
  0.630, 1.484, -0.001, 0.000,  0,  0,  0,  0, 1),
  nrow=23, ncol=9, byrow=TRUE)

ex1.data<-genFMPData(NSubj = NSubjects, bParams = b, seed = 345)$data

## number of items in the data matrix
NItems <- ncol(ex1.data)

# compute (initial) surrogate theta values from
# the normed left singular vector of the centered
# data matrix
thetaInit <- svdNorm(ex1.data)


## earlier we defined k = 1
  if(k == 0) {
           startVals <- c(1.5, 1.5)
```

```
            bmat <- matrix(0, NItems, 6)
          colnames(bmat) <- c(paste("b", 0:1, sep = ""),"FHAT", "AIC", "BIC", "convergence")
    }
    if(k == 1) {
             startVals <- c(1.5, 1.5, .10, .10)
             bmat <- matrix(0, NItems, 8)
           colnames(bmat) <- c(paste("b", 0:3, sep = ""),"FHAT", "AIC", "BIC", "convergence")
    }
    if(k == 2) {
             startVals <- c(1.5, 1.5, .10, .10, .10, .10)
             bmat <- matrix(0, NItems, 10)
           colnames(bmat) <- c(paste("b", 0:5, sep = ""),"FHAT", "AIC", "BIC", "convergence")
    }
    if(k == 3) {
             startVals <- c(1.5, 1.5, .10, .10, .10, .10, .10, .10)
             bmat <- matrix(0, NItems, 12)
           colnames(bmat) <- c(paste("b", 0:7, sep = ""),"FHAT", "AIC", "BIC", "convergence")
    }

  # estimate item parameters and fit statistics
    for(i in 1:NItems){
      out <- FMP(data = ex1.data, thetaInit, item = i, startvals = startVals, k = k)
      Nb <- length(out$b)
      bmat[i,1:Nb] <- out$b
      bmat[i,Nb+1] <- out$FHAT
      bmat[i,Nb+2] <- out$AIC
      bmat[i,Nb+3] <- out$BIC
      bmat[i,Nb+4] <- out$convergence
    }

  # print output
  print(bmat)

  ## End(Not run)
```

FMPMonotonicityCheck *Utility function for checking FMP monotonicity*

Description

Utility function for checking whether candidate FMP coefficients yield a monotonically increasing polynomial.

Usage

```
FMPMonotonicityCheck(b, lower = -20, upper = 20)
```

Arguments

b	A vector of 8 polynomial coefficients (b) for $m(\theta) = b_0 + b_1\theta + b_2\theta^2 + b_3\theta^3 + b_4\theta^4 + b_5\theta^5 + b_6\theta^6 + b_7\theta^7$.
lower, upper	Theta bounds for monotonicity check.

Value

minDeriv — Minimum value of the derivative for the polynomial.

Author(s)

Niels Waller

Examples

```
## A set of candidate coefficients for an FMP model.
## These coefficients fail the test and thus
## should not be used with genFMPdata to generate
## item response data that are consistent with an
## FMP model.
 b <- c(1.21, 1.87, -1.02, 0.18, 0.18, 0, 0, 0)
 FMPMonotonicityCheck(b)
```

FUP — *Estimate the coefficients of a filtered unconstrained polynomial IRT model*

Description

Estimate the coefficients of a filtered unconstrained polynomial IRT model.

Usage

```
FUP(data, thetaInit, item, startvals, k)
```

Arguments

data	N(subjects)-by-p(items) matrix of 0/1 item response data.
thetaInit	Initial theta surrogates (e.g., calculated by svdNorm).
item	item number for coefficient estimation.
startvals	start values for function minimization.
k	order of monotonic polynomial = 2k+1 (see Liang & Browne, 2015).

Value

b	Vector of polynomial coefficients.
FHAT	Function value at convergence.
counts	Number of function evaluations during minimization (see optim documentation for further details).
AIC	Pseudo scaled Akaike Information Criterion (AIC). Candidate models that produce the smallest AIC suggest the optimal number of parameters given the sample size. Scaling is accomplished by dividing the non-scaled AIC by sample size.

`BIC` Pseudo scaled Bayesian Information Criterion (BIC). Candidate models that produce the smallest BIC suggest the optimal number of parameters given the sample size. Scaling is accomplished by dividing the non-scaled BIC by sample size.

`convergence` Convergence = 0 indicates that the optimization algorithm converged; convergence=1 indicates that the optimization failed to converge.

.

Author(s)

Niels Waller

References

Liang, L. & Browne, M. W. (2015). A quasi-parametric method for fitting flexible item response functions. *Journal of Educational and Behavioral Statistics, 40*, 5–34.

Examples

```
## Not run:
NSubjects <- 2000


## generate sample k=1 FMP data
b <- matrix(c(
    #b0    b1     b2    b3      b4   b5 b6 b7  k
  1.675, 1.974, -0.068, 0.053,  0,  0,  0,  0, 1,
  1.550, 1.805, -0.230, 0.032,  0,  0,  0,  0, 1,
  1.282, 1.063, -0.103, 0.003,  0,  0,  0,  0, 1,
  0.704, 1.376, -0.107, 0.040,  0,  0,  0,  0, 1,
  1.417, 1.413,  0.021, 0.000,  0,  0,  0,  0, 1,
 -0.008, 1.349, -0.195, 0.144,  0,  0,  0,  0, 1,
  0.512, 1.538, -0.089, 0.082,  0,  0,  0,  0, 1,
  0.122, 0.601, -0.082, 0.119,  0,  0,  0,  0, 1,
  1.801, 1.211,  0.015, 0.000,  0,  0,  0,  0, 1,
 -0.207, 1.191,  0.066, 0.033,  0,  0,  0,  0, 1,
 -0.215, 1.291, -0.087, 0.029,  0,  0,  0,  0, 1,
  0.259, 0.875,  0.177, 0.072,  0,  0,  0,  0, 1,
 -0.423, 0.942,  0.064, 0.094,  0,  0,  0,  0, 1,
  0.113, 0.795,  0.124, 0.110,  0,  0,  0,  0, 1,
  1.030, 1.525,  0.200, 0.076,  0,  0,  0,  0, 1,
  0.140, 1.209,  0.082, 0.148,  0,  0,  0,  0, 1,
  0.429, 1.480, -0.008, 0.061,  0,  0,  0,  0, 1,
  0.089, 0.785, -0.065, 0.018,  0,  0,  0,  0, 1,
 -0.516, 1.013,  0.016, 0.023,  0,  0,  0,  0, 1,
  0.143, 1.315, -0.011, 0.136,  0,  0,  0,  0, 1,
  0.347, 0.733, -0.121, 0.041,  0,  0,  0,  0, 1,
 -0.074, 0.869,  0.013, 0.026,  0,  0,  0,  0, 1,
  0.630, 1.484, -0.001, 0.000,  0,  0,  0,  0, 1),
  nrow=23, ncol=9, byrow=TRUE)

# generate data using the above item parameters
ex1.data<-genFMPData(NSubj = NSubjects, bParams = b, seed = 345)$data

NItems <- ncol(ex1.data)
```

```
# compute (initial) surrogate theta values from
# the normed left singular vector of the centered
# data matrix
thetaInit <- svdNorm(ex1.data)

# Choose model
k <- 1  # order of polynomial = 2k+1

# Initialize matrices to hold output
if(k == 0) {
  startVals <- c(1.5, 1.5)
  bmat <- matrix(0,NItems,6)
  colnames(bmat) <- c(paste("b", 0:1, sep = ""),"FHAT", "AIC", "BIC", "convergence")
}

if(k == 1) {
  startVals <- c(1.5, 1.5, .10, .10)
  bmat <- matrix(0,NItems,8)
  colnames(bmat) <- c(paste("b", 0:3, sep = ""),"FHAT", "AIC", "BIC", "convergence")
}

if(k == 2) {
  startVals <- c(1.5, 1.5, .10, .10, .10, .10)
  bmat <- matrix(0,NItems,10)
  colnames(bmat) <- c(paste("b", 0:5, sep = ""),"FHAT", "AIC", "BIC", "convergence")
}

if(k == 3) {
  startVals <- c(1.5, 1.5, .10, .10, .10, .10, .10, .10)
  bmat <- matrix(0,NItems,12)
  colnames(bmat) <- c(paste("b", 0:7, sep = ""),"FHAT", "AIC", "BIC", "convergence")
}


# estimate item parameters and fit statistics
for(i in 1:NItems){
 out<-FUP(data = ex1.data,thetaInit = thetaInit, item = i, startvals = startVals, k = k)
  Nb <- length(out$b)
  bmat[i,1:Nb] <- out$b
  bmat[i,Nb+1] <- out$FHAT
  bmat[i,Nb+2] <- out$AIC
  bmat[i,Nb+3] <- out$BIC
  bmat[i,Nb+4] <- out$convergence
}

# print results
print(bmat)

## End(Not run)
```

---

gen4PMData *Generate item response data for 1, 2, 3, or 4-parameter IRT models*

Description

Generate item response data for or 1, 2, 3 or 4-parameter IRT Models.

Usage

```
gen4PMData(NSubj, abcdParams, D = 1.702,  seed = NULL,
                     theta = NULL, thetaMN = 0, thetaVar = 1)
```

Arguments

NSubj	the desired number of subject response vectors.
abcdParams	a p(items)-by-4 matrix of IRT item parameters: a = discrimination, b = difficulty, c = lower asymptote, and d = upper asymptote.
D	Scaling constant to place the IRF on the normal ogive or logistic metric. Default = 1.702 (normal ogive metric)
seed	Optional seed for the random number generator.
theta	Optional vector of latent trait scores. If theta = NULL (the default value) then gen4PMData will simulate theta from a normal distribution.
thetaMN	Mean of simulated theta distribution. Default = 0.
thetaVar	Variance of simulated theta distribution. Default = 1

Value

data	N(subject)-by-p(items) matrix of item response data.
theta	Latent trait scores.
seed	Value of the random number seed.

Author(s)

Niels Waller

Examples

```
## Generate simulated 4PM data for 2,000 subjects
# 4PM Item parameters from MMPI-A CYN scale

Params<-matrix(c(1.41, -0.79, .01, .98, #1
                 1.19, -0.81, .02, .96, #2
                 0.79, -1.11, .05, .94, #3
                 0.94, -0.53, .02, .93, #4
                 0.90, -1.02, .04, .95, #5
                 1.00, -0.21, .02, .84, #6
                 1.05, -0.27, .02, .97, #7
                 0.90, -0.75, .04, .73, #8
                 0.80, -1.42, .06, .98, #9
                 0.71,  0.13, .05, .94, #10
                 1.01, -0.14, .02, .81, #11
                 0.63,  0.18, .18, .97, #12
                 0.68,  0.18, .02, .87, #13
                 0.60, -0.14, .09, .96, #14
                 0.85, -0.71, .04, .99, #15
                 0.83, -0.07, .05, .97, #16
                 0.86, -0.36, .03, .95, #17
                 0.66, -0.64, .04, .77, #18
                 0.60,  0.52, .04, .94, #19
```

```
                0.90, -0.06, .02, .96, #20
                0.62, -0.47, .05, .86, #21
                0.57,  0.13, .06, .93, #22
                0.77, -0.43, .04, .97),23,4, byrow=TRUE)

data <- gen4PMData(NSubj=2000, abcdParams = Params, D = 1.702,
                   seed = 123, thetaMN = 0, thetaVar = 1)$data

cat("\nClassical item difficulties for simulated data")
print( round( apply(data,2,mean),2) )
```

`genFMPData` *Generate item response data for a filtered monotonic polynomial IRT model*

Description

Generate item response data for the filtered polynomial IRT model.

Usage

```
genFMPData(NSubj, bParams, theta = NULL, thetaMN = 0, thetaVar = 1, seed)
```

Arguments

`NSubj`	the desired number of subject response vectors.
`bParams`	a p(items)-by-9 matrix of polynomial coefficients and model designations. Columns 1 - 8 hold the polynomial coefficients; column 9 holds the value of `k`.
`theta`	A user-supplied vector of latent trait scores. Default theta = NULL.
`thetaMN`	If theta = NULL genFMPdata will simulate random normal deviates from a population with mean thetaMN and variance thetaVar.
`thetaVar`	If theta = NULL genFMPData will simulate random normal deviates from a population with mean thetaMN and variance thetaVar.
`seed`	initial seed for the random number generator.

Value

`theta`	theta values used for data generation
`data`	N(subject)-by-p(items) matrix of item response data.
`seed`	Value of the random number seed.

Author(s)

Niels Waller

Examples

```
# The following code illustrates data generation for
# an FMP of order 3 (i.e., 2k+1)

# data will be generated for 2000 examinees
NSubjects <- 2000


## Example item paramters, k=1 FMP
b <- matrix(c(
    #b0    b1     b2    b3      b4   b5 b6 b7  k
  1.675, 1.974, -0.068, 0.053,  0,  0,  0,  0, 1,
  1.550, 1.805, -0.230, 0.032,  0,  0,  0,  0, 1,
  1.282, 1.063, -0.103, 0.003,  0,  0,  0,  0, 1,
  0.704, 1.376, -0.107, 0.040,  0,  0,  0,  0, 1,
  1.417, 1.413,  0.021, 0.000,  0,  0,  0,  0, 1,
 -0.008, 1.349, -0.195, 0.144,  0,  0,  0,  0, 1,
  0.512, 1.538, -0.089, 0.082,  0,  0,  0,  0, 1,
  0.122, 0.601, -0.082, 0.119,  0,  0,  0,  0, 1,
  1.801, 1.211,  0.015, 0.000,  0,  0,  0,  0, 1,
 -0.207, 1.191,  0.066, 0.033,  0,  0,  0,  0, 1,
 -0.215, 1.291, -0.087, 0.029,  0,  0,  0,  0, 1,
  0.259, 0.875,  0.177, 0.072,  0,  0,  0,  0, 1,
 -0.423, 0.942,  0.064, 0.094,  0,  0,  0,  0, 1,
  0.113, 0.795,  0.124, 0.110,  0,  0,  0,  0, 1,
  1.030, 1.525,  0.200, 0.076,  0,  0,  0,  0, 1,
  0.140, 1.209,  0.082, 0.148,  0,  0,  0,  0, 1,
  0.429, 1.480, -0.008, 0.061,  0,  0,  0,  0, 1,
  0.089, 0.785, -0.065, 0.018,  0,  0,  0,  0, 1,
 -0.516, 1.013,  0.016, 0.023,  0,  0,  0,  0, 1,
  0.143, 1.315, -0.011, 0.136,  0,  0,  0,  0, 1,
  0.347, 0.733, -0.121, 0.041,  0,  0,  0,  0, 1,
 -0.074, 0.869,  0.013, 0.026,  0,  0,  0,  0, 1,
  0.630, 1.484, -0.001, 0.000,  0,  0,  0,  0, 1),
  nrow=23, ncol=9, byrow=TRUE)

# generate data using the above item paramters
data<-genFMPData(NSubj = NSubjects, bParams=b, seed=345)$data
```

`irf` *Plot item response functions for polynomial IRT models.*

Description

Plot model-implied (and possibly empirical) item response function for polynomial IRT models.

Usage

```
irf(data, bParams, item, plotERF = TRUE, thetaEAP = NULL,
                minCut = -3, maxCut = 3, NCuts = 9)
```

Arguments

`data`	N(subjects)-by-p(items) matrix of 0/1 item response data.
`bParams`	p(items)-by-9 matrix. The first 8 columns of the matrix should contain the FMP or FUP polynomial coefficients for the p items. The 9th column contains the value of k for each item (where the item specific order of the polynomial is 2k+1).
`item`	The IRF for `item` will be plotted.
`plotERF`	A logical that determines whether to plot discrete values of the empirical response function.
`thetaEAP`	If `plotERF=TRUE`, the user must supply previously calculated eap trait estimates to `thetaEAP`.
`minCut, maxCut`	If `plotERF=TRUE`, the program will (attempt to) plot `NCuts` points of the empirical response function between trait values of `minCut` and `maxCut` Default minCut = -3. Default maxCut = 3.
`NCuts`	Desired number of bins for the empirical response function.

Author(s)

Niels Waller

Examples

```
NSubjects <- 2000
NItems <- 15

itmParameters <- matrix(c(
 #  b0    b1     b2    b3    b4  b5,    b6,  b7,  k
 -1.05, 1.63,  0.00, 0.00, 0.00,  0,     0,  0,   0, #1
 -1.97, 1.75,  0.00, 0.00, 0.00,  0,     0,  0,   0, #2
 -1.77, 1.82,  0.00, 0.00, 0.00,  0,     0,  0,   0, #3
 -4.76, 2.67,  0.00, 0.00, 0.00,  0,     0,  0,   0, #4
 -2.15, 1.93,  0.00, 0.00, 0.00,  0,     0,  0,   0, #5
 -1.25, 1.17, -0.25, 0.12, 0.00,  0,     0,  0,   1, #6
  1.65, 0.01,  0.02, 0.03, 0.00,  0,     0,  0,   1, #7
 -2.99, 1.64,  0.17, 0.03, 0.00,  0,     0,  0,   1, #8
 -3.22, 2.40, -0.12, 0.10, 0.00,  0,     0,  0,   1, #9
 -0.75, 1.09, -0.39, 0.31, 0.00,  0,     0,  0,   1, #10
 -1.21, 9.07,  1.20,-0.01,-0.01,  0.01,  0,  0,   2, #11
 -1.92, 1.55, -0.17, 0.50,-0.01,  0.01,  0,  0,   2, #12
 -1.76, 1.29, -0.13, 1.60,-0.01,  0.01,  0,  0,   2, #13
 -2.32, 1.40,  0.55, 0.05,-0.01,  0.01,  0,  0,   2, #14
 -1.24, 2.48, -0.65, 0.60,-0.01,  0.01,  0,  0,   2),#15
 15, 9, byrow=TRUE)


ex1.data<-genFMPData(NSubj = NSubjects, bParams = itmParameters,
                     seed = 345)$data

## compute initial theta surrogates
thetaInit <- svdNorm(ex1.data)

## For convenience we assume that the item parameter
## estimates equal their population values.  In practice,
```

```
## item parameters would be estimated at this step.
itmEstimates <- itmParameters

## calculate eap estimates for mixed models
thetaEAP <- eap(data = ex1.data, bParams = itmEstimates, NQuad = 21,
                priorVar = 2,
                mintheta = -4, maxtheta = 4)

## plot irf and erf for item 1
irf(data = ex1.data, bParams = itmEstimates,
    item = 1,
    plotERF = TRUE,
    thetaEAP)

## plot irf and erf for item 12
irf(data = ex1.data, bParams = itmEstimates,
    item = 12,
    plotERF = TRUE,
    thetaEAP)
```

---

restScore *Plot an ERF using rest scores*

Description

Plot an empirical response function using rest scores.

Usage

```
restScore(data, item, NCuts)
```

Arguments

`data`	N(subjects)-by-p(items) matrix of 0/1 item response data.
`item`	Generate a rest score plot for item `item`.
`NCuts`	Divide the rest scores into `NCuts` bins of equal width.

Value

A restscore plot with 95% confidence interval bars for the conditional probability estimates.

`item`	The item number.
`bins`	A vector of bin limits and bin sample sizes.
`binProb`	A vector of bin conditional probabilities.

Author(s)

Niels Waller

Examples

```
NSubj <- 2000

#generate sample k=1 FMP  data
b <- matrix(c(
    #b0    b1     b2    b3      b4   b5  b6  b7   k
  1.675, 1.974, -0.068, 0.053,  0,  0,  0,  0, 1,
  1.550, 1.805, -0.230, 0.032,  0,  0,  0,  0, 1,
  1.282, 1.063, -0.103, 0.003,  0,  0,  0,  0, 1,
  0.704, 1.376, -0.107, 0.040,  0,  0,  0,  0, 1,
  1.417, 1.413,  0.021, 0.000,  0,  0,  0,  0, 1,
 -0.008, 1.349, -0.195, 0.144,  0,  0,  0,  0, 1,
  0.512, 1.538, -0.089, 0.082,  0,  0,  0,  0, 1,
  0.122, 0.601, -0.082, 0.119,  0,  0,  0,  0, 1,
  1.801, 1.211,  0.015, 0.000,  0,  0,  0,  0, 1,
 -0.207, 1.191,  0.066, 0.033,  0,  0,  0,  0, 1,
 -0.215, 1.291, -0.087, 0.029,  0,  0,  0,  0, 1,
  0.259, 0.875,  0.177, 0.072,  0,  0,  0,  0, 1,
 -0.423, 0.942,  0.064, 0.094,  0,  0,  0,  0, 1,
  0.113, 0.795,  0.124, 0.110,  0,  0,  0,  0, 1,
  1.030, 1.525,  0.200, 0.076,  0,  0,  0,  0, 1,
  0.140, 1.209,  0.082, 0.148,  0,  0,  0,  0, 1,
  0.429, 1.480, -0.008, 0.061,  0,  0,  0,  0, 1,
  0.089, 0.785, -0.065, 0.018,  0,  0,  0,  0, 1,
 -0.516, 1.013,  0.016, 0.023,  0,  0,  0,  0, 1,
  0.143, 1.315, -0.011, 0.136,  0,  0,  0,  0, 1,
  0.347, 0.733, -0.121, 0.041,  0,  0,  0,  0, 1,
 -0.074, 0.869,  0.013, 0.026,  0,  0,  0,  0, 1,
  0.630, 1.484, -0.001, 0.000,  0,  0,  0,  0, 1),
  nrow=23, ncol=9, byrow=TRUE)

data<-genFMPData(NSubj = NSubj, bParam = b, seed = 345)$data

## generate a rest score plot for item 12.
## the grey horizontal lines in the plot
## respresent pseudo asymptotes that
## are significantly different from the
## (0,1) boundaries
restScore(data, item = 12, NCuts = 9)
```

---

svdNorm — *Compute theta surrogates via normalized SVD scores*

Description

Compute theta surrogates by calculating the normalized left singular vector of a (mean-centered) data matrix.

Usage

```
svdNorm(data)
```

Arguments

`data`	N(subjects)-by-p(items) matrix of 0/1 item response data.

Value

the normalized left singular vector of the mean centered data matrix.
svdNorm will center the data automatically.

Author(s)

Niels Waller

Examples

```
NSubj <- 2000

## example item parameters for sample data: k=1 FMP
b <- matrix(c(
    #b0    b1     b2    b3      b4   b5 b6 b7  k
  1.675, 1.974, -0.068, 0.053,  0,  0,  0,  0, 1,
  1.550, 1.805, -0.230, 0.032,  0,  0,  0,  0, 1,
  1.282, 1.063, -0.103, 0.003,  0,  0,  0,  0, 1,
  0.704, 1.376, -0.107, 0.040,  0,  0,  0,  0, 1,
  1.417, 1.413,  0.021, 0.000,  0,  0,  0,  0, 1,
 -0.008, 1.349, -0.195, 0.144,  0,  0,  0,  0, 1,
  0.512, 1.538, -0.089, 0.082,  0,  0,  0,  0, 1,
  0.122, 0.601, -0.082, 0.119,  0,  0,  0,  0, 1,
  1.801, 1.211,  0.015, 0.000,  0,  0,  0,  0, 1,
 -0.207, 1.191,  0.066, 0.033,  0,  0,  0,  0, 1,
 -0.215, 1.291, -0.087, 0.029,  0,  0,  0,  0, 1,
  0.259, 0.875,  0.177, 0.072,  0,  0,  0,  0, 1,
 -0.423, 0.942,  0.064, 0.094,  0,  0,  0,  0, 1,
  0.113, 0.795,  0.124, 0.110,  0,  0,  0,  0, 1,
  1.030, 1.525,  0.200, 0.076,  0,  0,  0,  0, 1,
  0.140, 1.209,  0.082, 0.148,  0,  0,  0,  0, 1,
  0.429, 1.480, -0.008, 0.061,  0,  0,  0,  0, 1,
  0.089, 0.785, -0.065, 0.018,  0,  0,  0,  0, 1,
 -0.516, 1.013,  0.016, 0.023,  0,  0,  0,  0, 1,
  0.143, 1.315, -0.011, 0.136,  0,  0,  0,  0, 1,
  0.347, 0.733, -0.121, 0.041,  0,  0,  0,  0, 1,
 -0.074, 0.869,  0.013, 0.026,  0,  0,  0,  0, 1,
  0.630, 1.484, -0.001, 0.000,  0,  0,  0,  0, 1),
  nrow=23, ncol=9, byrow=TRUE)

# generate data using the above item paramters
data<-genFMPData(NSubj=NSubj, bParam=b, seed=345)$data

# compute (initial) surrogate theta values from
# the normed left singular vector of the centered
# data matrix
thetaInit<-svdNorm(data)
```

Index