Package: optical (via r-universe)

October 8, 2024

Type Package

Title Optimal Item Calibration

Version 1.8

Date 2024-01-01

Depends R (>= 4.1.0)

Description The restricted optimal design method is implemented to optimally allocate a set of items that require calibration to a group of examinees. The optimization process is based on the method described in detail by Ul Hassan and Miller in their works published in (2019) <doi:10.1007/s11336-019-09673-6> and (2021) <doi:10.1016/j.csda.2021.107177>. To use the method, preliminary item characteristics must be provided as input. These characteristics can either be expert guesses or based on previous calibration with a small number of examinees. The item characteristics should be described in the form of parameters for an Item Response Theory (IRT) model. These models can include the Rasch model, the 2-parameter logistic model, the 3-parameter logistic model, or a mixture of these models. The output consists of a set of rules for each item that determine which examinees should be assigned to each item. The efficiency or gain achieved through the optimal design is quantified by comparing it to a random allocation. This comparison allows for an assessment of how much improvement or advantage is gained by using the optimal design approach. This work was supported by the Swedish Research Council (Vetenskapsrådet) Grant 2019-02706.

Author Mahmood Ul Hassan [aut, cre] (<https://orcid.org/0000-0003-2889-0263>), Frank Miller [aut] (<https://orcid.org/0000-0003-4161-7851>)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

License GPL (>= 3)

License_restricts_use no

Maintainer Mahmood Ul Hassan <scenic555@gmail.com>

BugReports https://github.com/scenic555/optical/issues

URL https://scenic555.github.io/optical/,
https://github.com/scenic555/optical

Language en-GB

Imports stats

Repository https://scenic555.r-universe.dev

RemoteUrl https://github.com/scenic555/optical

RemoteRef HEAD

RemoteSha 4774e25545d8afd7ca3bb0560902e0fdb91e8e66

Contents

---

convergenceplot *Convergence plot*

Description

For each block, convergence plots display efficiency of design, violations of the equivalence theorem, the step length used vs. iteration number. These plots are suitable for monitoring the convergence of the optimal item calibration algorithm.

Usage

```
convergenceplot(yyy, refline = c(0.002, 1e-05))
```

Arguments

yyy	an optical object; the output of a call optical().
refline	reference line.

Details

Convergence plots have three panels.

- First panel monitors efficiency of design vs. iteration number.
- Second panel monitors violations of equivalence theorem vs. iteration number.
- Third panel monitors step size used vs. iteration number.

Value

A convergence plot is displayed.

See Also

drawdesign, drawdesign_allitems

Examples

```
# Example No.1
# 2PL-model for three items with parameters (a, b) equal to (1.6, -2), (1.6, 0.5),
# and (1.6, 2) for the first, second, and third items, respectively.

ip <- cbind(c(1.6, 1.6,1.6),c(-2, 0.5,2))
yyy <- optical(ip)
convergenceplot(yyy, refline=c(0.002, 0.001*0.005/0.45))

## Not run:

# Example No.2
# 2PL-models for six items; parameters a=(1.62, 1.4, 0.98, 0.66, 0.92, 0.9),
# and b=(-0.47, -1.71, 0.62, -0.15, -1.71, 1.6), respectively.

a<-c(1.62, 1.4,0.98,0.66,0.92,0.9)
b<-c(-0.47,-1.71,0.62,-0.15,-1.71,1.6)
ip<-cbind(a,b)
bid<-c(1,1,1,2,2,2)
yyy <- optical(ip,bid=bid,show_progress=2)
convergenceplot(yyy, refline=c(0.002, 0.001*0.005/0.45))


## End(Not run)
```

---

drawdesign *Graph for (optimal) design in each block*

Description

Generates a plot for the optimal design within each block, showcasing six possible layouts. All layouts feature the design first, including efficiency versus random design, followed by a line representing the item with the minimal directional derivative.

Usage

```
drawdesign(
  yyy,
  ablim = 7,
  ylowl = -9999999,
  refline = 0.002,
  textout = TRUE,
  itemnum = NA,
  layout = 1,
  colvec = 1:12
)
```

Arguments

`yyy`	an `optical` object; the output of a call `optical()`.
`ablim`	ability limits; plots will be made in the range [-ablim, ablim].
`ylowl`	y low level (minimum value of directional derivative shown in the plot).
`refline`	reference line corresponding to desired minimum violation of equivalence theorem.
`textout`	If textout=TRUE (default), the item parameters will be printed if number of items <5 and the efficiency vs. the random design; if textout=FALSE, no such text is printed.
`itemnum`	number of items.
`layout`	layouts of plots. • Layout 1: third panel has directional derivatives (cut at ylowl or lowest value of dirdev). • Layout 2: third panel has violations of equivalence theorem, should be ideally small. Stopping criterion could be <0.002 (refline). • Layout 3: third panel monitors efficiency of design vs. iteration number. • Layout 4: third panel monitors violations of equivalence theorem vs. iteration number. • Layout 5: third panel shows item characteristic curves. • Layout 0: only one panel with design.
`colvec`	vector of color sequence for items (default is the R-default black, red, green, etc.)

Value

An optimal design plot is displayed.

See Also

`convergenceplot`

Examples

```
# Example No.1
# 2PL-model for three items with parameters (a, b) equal to (1.6, -2), (1.6, 0.5),
# and (1.6, 2) for the first, second, and third items, respectively.

a<-c(1.6,1.6,1.6); b<-c(-2,0.5,2)
ip <- cbind(a,b)
yyy <- optical(ip)
drawdesign(yyy, ylowl=-1000, refline=0.002)

## Not run:

# Example No.2
# 2PL-models for six items; the parameters for these items are a=(1.62, 1.4, 0.98, 0.66, 0.92, 0.9),
# and b=(-0.47, -1.71, 0.62, -0.15, -1.71, 1.6), respectively.

a<-c(1.62, 1.4,0.98,0.66,0.92,0.9); b<-c(-0.47,-1.71, 0.62,-0.15,-1.71,1.6)
ip<-cbind(a,b)
bid<-c(1,1,1,2,2,2)
yyy <- optical(ip,BID=bid,show_progress=2)

drawdesign(yyy, ylowl=-1000, refline=0.002, layout=5)


## End(Not run)
```

---

drawdesign_allitems *Optimal calibration design for all items*

Description

Create a plot with optimal design for each item. Optimal design is represented by different colored lines for each item.

Usage

```
drawdesign_allitems(yyy, linew = NULL, ablim = 4, colvec = NULL)
```

Arguments

yyy	an `optical` object; the output of a call `optical()`.
linew	linewidth for design.
ablim	ability limits; plots will be made in the range [-ablim, ablim].
colvec	vector of color sequence for items (default is the R-default black, red, green, etc.).

Value

An optimal design plot is displayed for all items.

Examples

```
# Example No.1
# 2PL-model for three items with parameters (a, b) equal to (1.6, -2), (1.6, 0.5),
# and (1.6, 2) for the first, second, and third items, respectively.

ip <- cbind(c(1.6, 1.6, 1.6), c(-2, 0.5, 2))
yyy <- optical(ip)
drawdesign_allitems(yyy)


## Not run:
# Example No.2
# 2PL-models for six items; the parameters for these items are a=(1.62, 1.4, 0.98, 0.66, 0.92, 0.9),
# and b=(-0.47, -1.71, 0.62, -0.15, -1.71, 1.6), respectively.

ip <- cbind(c(1.62, 1.4, 0.98, 0.66, 0.92, 0.9), c(-0.47, -1.71, 0.62, -0.15, -1.71, 1.6))
bid <- c(1, 1, 1, 2, 2, 2)
yyy <- optical(ip, bid = bid, show_progress = 2)
drawdesign_allitems(yyy)

# Items of the same color that are in the same block.
drawdesign_allitems(yyy, colvec = bid)

## End(Not run)
```

`efficiency` *Efficiency of optimal design*

Description

This function computes the efficiency of the D-, I-, and A-optimal designs compared to the random design for each block.

Usage

```
efficiency(
  yyy,
  uncert = FALSE,
  ipop,
  oc = "D",
  L = NULL,
  items = FALSE,
  integ = TRUE
)
```

Arguments

yyy	a `optical` object; the output of a call `optical()`.

uncert	if false (default), abilities are assumed to be known; if true, handling of uncertainties of Bjermo et al. (2021) is used.
ipop	matrix with item parameters for operational items (used if uncert=TRUE, only).
oc	optimality criterion: "D" (D-optimality, default), "I" (I-optimality with standard normal weight function), "A" (A-optimality).
L	L-matrix (not used for D-optimality).
items	If false (default), only total block efficiency is returned. If true, for each block, criteria for optimal and random, and the efficiency for each item are reported in each column of output, except for 1-pl model where each column represents the parameter efficiency. The last column shows total criteria and efficiency. D-, L-, I-, A-optimality.
integ	if true (default), integrate() is used for computation of partial information matrices; if false, Riemann rule is used.

Value

A numerical value is displayed.

See Also

optical

Examples

```
# Example No.1
# 2PL-models for two items; parameters (a, b)=(1.6, -1) and (1.6, 1), respectively

a<-c(1.6, 1.6); b<-c(-1, 1)
ip <- cbind(a,b)
yyy <- optical(ip)

# Efficiency of A-optimal design compared to random design
efficiency(yyy, oc="A")

## Not run:

# Example No.2
# 2PL-models for six items; the parameters for these items are a=(1.62, 1.4, 0.98, 0.66, 0.92, 0.9),
# and b=(-0.47, -1.71, 0.62, -0.15, -1.71, 1.6), respectively.

a <- c(1.62, 1.4, 0.98, 0.66, 0.92, 0.9)
b <- c(-0.47, -1.71, 0.62, -0.15, -1.71, 1.6)
ip <- cbind(a, b)
bid <- c(1, 1, 1, 2, 2, 2)
yyy <- optical(ip, bid = bid, show_progress = 2)

# Efficiency of D-optimal design compared to random design
efficiency(yyy, oc = "D")

# Efficiency of A-optimal design compared to random design
```

```
efficiency(yyy, oc = "A")

# Efficiency of I-optimal design compared to random design
efficiency(yyy, oc = "I")

## End(Not run)
```

---

optical *Optimal item calibration*

Description

Calibrate items using a 2PL, 3PL, a mixture of 2PL and 3PL models, or a 2PL model with a common discrimination parameter for all items (we call this Rasch-type). This versatile function is designed to address calibration challenges involving a substantial number of items (100-200) by grouping them into blocks of 3 to 5 items each.

Usage

```
optical(
  ip,
  bid = NULL,
  oc = "D",
  uncert = FALSE,
  ipop,
  imf = c(0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.45),
  maxiter = rep(300, 6),
  eps = rep(0.002, 6),
  nnn = c(0, 50, 50, 200, 200, 200),
  nsp = c(0.001, 1e-04, 1e-04, 1e-05, 1e-05, 1e-05),
  sss = 0.001,
  falpha = 1.08,
  sdr = TRUE,
  ig = 3,
  ex = 0,
  integ = TRUE,
  show_progress = 1
)
```

Arguments

`ip` Matrix containing item parameters for all items. The number of rows corresponds to the number of items, and columns represent item parameters. There are four possible configurations:

- 1. Rasch-type: The first two columns contain item parameters, while the first column contains 'NA' from the second item.
- 2. 2PL items: First two columns contain item parameters.

- 3. 3PL items: All three columns contain item parameters.
- 4. Mixed 2/3-PL: For 3-PL items, all three columns contain item parameters, while 2-PL items have NA in the third column.

`bid` Vector of integers indicating the block assignments for items. The length of bid should match the number of rows in the 'ip' matrix. Each element in the vector specifies which item belongs to which block. bid is utilized in the calibration process when dealing with more than 5 items. Note that the block should exclusively consist of Rasch-type items, 2-PL items, 3-PL items, or a combination of 2-3PL items. If blocks consist of Rasch-type, 2-PL, and 3PL items, then we need to introduce a third column with 'NA' for 2-PL and Rasch-type items in ip matrix.

`oc` optimality criterion: "D" (D-optimality, default), "I" (I-optimality with standard normal weight function), "A" (A-optimality).

`uncert` if false (default), abilities are assumed to be known; if true, handling of uncertainties of Bjermo et al. (2021) is used.

`ipop` matrix with item parameters for operational items (used if uncert=TRUE, only).

`imf` the vector of step-lengths; default c(0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.45).

`maxiter` maximal number of iterations in each inner loop, the length of this vector defines the number of outer loops.

`eps` convergence criterion (maximum violation of eq.th.), vector with value for each iteration in the outer loop, but the same number for all iterations is recommended.

`nnn` number of new nodes added at each position in the adaptive grid, vector with value for each iteration in the outer loop (nnn [1] not used).

`nsp` node spacing between new nodes, vector with value for each iteration in the outer loop (nsp [1] is the spacing between nodes of the starting grid).

`sss` step size stopping criterion.

`falpha` factor alpha for adjusting the step size vector (should be > 1).

`sdr` stop if design repeated (flag TRUE/FALSE).

`ig` inner grid between -ig and ig.

`ex` intervals of size < ex will be removed (consolidate); if ex=0, no consolidation will be done.

`integ` if true (default), integrate() is used for computation of partial information matrices; if false, Riemann rule is used.

`show_progress` if 1 (default), no output will be printed for each iteration. If 2, the + symbols will be printed on a line for each Iteration If 3, some output of the function will be printed.

Value

an object of class 'optical' is returned, which is a list with following instances:

`dd` list of directional derivatives of optimal solution for each block.

ip	list having matrix of item paramters in each block.
xi	list of optimal solution for each block.
t	list of final grid of ability values which was used in each block.
viomax	largest violation of eq.th. from final solution (if < eps, alg. has converged, otherwise not).
h1	list of interval boundaries for optimal solution for each block.
ht	list of Refined table of interval boundaries for optimal design with calibrated items and their corresponding probabilities for each block.
mooiter	list for each block having monitoring iterations; information about each iteration to produce convergence plots.
time	running time of algorithm in minutes.
oc	list of optimality criterion ("D", "I", "A", "L") for each block.
L	list of L-matrix (not for D-optimality) for each block.

Author(s)

Mahmood Ul Hassan (<scenic555@gmail.com>); Frank Miller (<frank.miller@liu.se>)

References

Ul Hassan and Miller (2021). An exchange algorithm for optimal calibration of items in computerized achievement tests.*Computational Statistics and Data Analysis, 157*: 107177.

Ul Hassan and Miller (2019). Optimal item calibration for computerized achievement tests. Psychometrika, 84, 1101-1128.

Bjermo, Fackle-Fornius, and Miller (2021). Optimizing Calibration Designs with Uncertainty in Abilities. Manuscript.

See Also

drawdesign, convergenceplot, efficiency, drawdesign_allitems

Examples

```
# Example No.1
# 2PL-model for three items with parameters (a, b) equal to (1.6, -2), (1.6, 0.5),
# and (1.6, 2) for the first, second, and third items, respectively.
# The calibration of these three items with the 2PL model is conducted in one block.

a<-c(1.6,1.6,1.6); b<-c(-2,0.5,2)
ip <-cbind(a,b)
yyy <- optical(ip)
yyy

## Not run:

# Example No.2
# 2PL-models for six items; parameters a=c(1.62, 0.66, 0.92, 0.90, 0.98, 1.40)
```

```
# and b=c(-0.47, -0.15, -1.71, 1.60, 0.62, -1.71), respectively.
# The calibration of these 6 items with 2PL-models is conducted in two blocks.

a<-c(1.62, 0.66,0.92,0.90,0.98,1.40)
b<-c(-0.47,-0.15,-1.71,1.60,0.62,-1.71)
ip<-cbind(a,b)
bid<-c(1,2,2,2,1,1)
yyy<-optical(ip,bid=bid)
yyy

# Example No.3
# Two-parameter logistic (2PL) models were employed for nine items, with
# parameters set as follows: a=c(1.62, 0.66, 0.92, 0.82, 0.90, 0.98, 0.36, 1.40, 0.64)
# and b=c(-0.47,-0.15, -1.71, 0.33, 1.60, 0.62, 2.84, -1.71, -1.51), respectively.
# The calibration of these nine items is conducted using 2PL models in three blocks.

ip <- cbind(c(1.62,0.66,0.92,0.82,0.90,0.98,0.36,1.40,0.64),
            c(-0.47,-0.15,-1.71,0.33,1.60,0.62,2.84,-1.71,-1.51) )
bid<-c(1,2,2,3,2,1,1,3,3)
yyy <- optical(ip,bid=bid,show_progress=2)
yyy

# Example No.4
# 2PL-models with a common discrimination parameter for six items. (The model
# assumption is that six items have the same discrimination.) Parameters are
# a=(1.6, NA, NA, 1.2, NA, NA) and b=(-2, -1.5, -1, 0.5, 1, 1.5), respectively.
# 'NA' for discrimination means that the item has the same parameter as the
# preceding item. The calibration of these 6 items with 2PL-models is conducted
# in 3 blocks.

ip<-cbind(c(1.6, NA, NA, 1.2, NA, NA),c(-2, -1.5, -1, 0.5, 1, 1.5))
bid<-c(1,1,1,2,2,2)
yyy <- optical(ip,bid=bid,show_progress=2)
yyy

# Example No.5
# 3PL-models for three items; the parameters for these items are a=(1, 2, 2.5),
# b=(-1.5, 0.5, 2), and c=(0.2, 0.1, 0.05), respectively. The Calibration of
# items following 3PL-models is performed in one block.

a<-c(1, 2, 2.5); b<-c(-1.5, 0.5, 2); c<-c(0.2, 0.1, 0.05)
ip<-cbind(a,b,c)
yyy <- optical(ip,show_progress=2)
yyy

# Example No.6
# 1PL-models are applied for four items, 2PL-models to three items, 3PL-models
# to three items,and 2-3PL-models to four items. The calibration of items
# is performed in four block.

a<-c(c(1, 1.8, 1.4, 1, 2, NA, 1.7, NA, 2.5, NA, 0.8, 1.5, 2, 2.8))
b<-c(-1.5, -0.9, -0.1, -1.5, 0.5, -0.25, 1.1, 0.25, 2, 1.5, -0.5,-2, 0.5, 2)
c<-c(NA, NA, NA, 0.2, 0.1, NA, NA, NA, 0.05, NA, NA, NA, 0.1, 0.2)
```

```
ip<-cbind(a,b,c)
bid<-c(1, 2, 2, 3, 3, 1, 2, 1, 3, 1, 4, 4, 4, 4)
yyy <- optical(ip,show_progress=2)
yyy


## End(Not run)
```

---

print.optical *print an optical object*

Description

This function print the table of interval boundaries for optimal design with items and probabilities for each block.

Usage

```
## S3 method for class 'optical'
print(x,digits=max(3, getOption("digits")-3),...)
```

Arguments

x	optical object.
digits	significant digits in printout.
...	other print arguments.

Value

No return value, called for side effects.

See Also

optical

Index