Journal of Statistical Software

January 2017, Volume 76, Code Snippet 1. doi: 10.18637/jss.v076.c01

Computerized Adaptive Testing with R: Recent Updates of the Package catR

David Magis
University of Liège

Juan Ramon Barrada
Universidad de Zaragoza

Abstract

The purpose of this paper is to list the recent updates of the R package **catR**. This package allows for generating response patterns under a computerized adaptive testing (CAT) framework with underlying item response theory (IRT) models. Among the most important updates, well-known polytomous IRT models are now supported by **catR**; several item selection rules have been added; and it is now possible to perform post-hoc simulations. Some functions were also rewritten or withdrawn to improve the usefulness and performances of the package.

Keywords: computerized adaptive testing, polytomous IRT models, post-hoc simulations, R package.

1. Introduction

In the field of psychometrics, computerized adaptive testing (CAT) is an important area of current research and practical implementations have shown a huge increment in the last decade. Unlike traditional linear testing wherein all respondents receive the same set of items, the main purpose of CAT is to perform iterative and adaptive administration of the items. The items are selected and administered one by one, and the selection of the next item is conditional upon the previously administered items, the responses of the respondent and the provisional estimate of ability level. CAT has several advantages with respect to linear testing: Among others, it requires less items to reach the same level of precision for ability estimation, leading thus to shorter tests for the respondents, and ability estimates are available directly after the test administration for immediate feedback to the test takers.

Although the CAT literature has increased in the past two decades (e.g., van der Linden and Glas 2010; Wainer 2000), there is still a lack of open-source and flexible software to run CATs and to perform intensive simulation studies in this framework. The R (R Core Team

2016) package **catR** (Magis and Raîche 2012) was originally developed for this purpose. The package is available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=catR`. It offers a variety of options to generate response patterns in a CAT environment, by providing first a pre-calibrated item bank, then by selecting all options related to CAT assessments (such as ad-interim and final ability estimators, a method for next item selection and a stopping rule). Its general architecture makes **catR** flexible, easy to update, and several of its components can be used even outside the CAT framework (for instance, the ability estimation and related standard error computation routines). Though basically developed as a working routine for CAT studies, **catR** can also be used as the core computing for real CAT assessment platforms, such as the web-based platform **Concerto** (Kosinski *et al.* 2013).

Since its very first version 1.0, released in June 2010, the package was updated with minor yet important updates to fix programming errors and enhance general improvement, leading to version 2.6 (released in March 2013). Recently, **catR** received a major update, due to both an increasing interest for the package and the need for further developments to match more realistic situations. One major update was to incorporate most common polytomous item response theory (IRT) models into **catR**. This mandatory extension was motivated by the fact that most questionnaires contain polytomous (e.g. multiple-choice) items for which specific models exist but were not yet available in **catR**.

The purpose of this note is to briefly review the major changes and improvements of **catR** from version 2.6 to its most recent version 3.12 (released in January 2017). Sections 2 to 4 present the three main updates of **catR**: the inclusion of polytomous IRT models, the implementation of additional item selection rules, and the option to run post-hoc simulations. Several technical details are also included in Appendix A. The package itself will not be described again, so we refer the interested reader to Magis and Raîche (2012) for more details.

2. Polytomous IRT models

As already mentioned, the main update of **catR** involves the inclusion of the most common polytomous IRT models: the *graded response model* (GRM; Samejima 1969, 1996), the *modified graded response model* (MGRM; Muraki 1990), the *partial credit model* (PCM; Masters 1982), the *generalized partial credit model* (GPCM; Muraki 1992), the *rating scale model* (RSM; Andrich 1978a,b) and the *nominal response model* (NRM; Bock 1972). These models were integrated into the package with the following requirements and guidelines: (a) **catR** function names were not modified; (b) by default, all functions remain operational with dichotomous IRT models; (c) all functions support polytomous IRT models and return similar yet appropriate output. These choices were made to prevent a deep modification of the current use of **catR**, especially for researchers who are currently using the package with dichotomous IRT models.

The specification of a polytomous IRT model is composed of two elements: an appropriately defined matrix of item parameters and the new argument `model` added to almost all existing functions. By default, `model` takes the `NULL` value and refers to dichotomous models (for which the item bank format is left unchanged from previous versions of **catR**). Other possible values are the polytomous model acronyms, for instance `"GRM"` for the graded response model, `"PCM"` for the partial credit model and so on.

The format of a bank of item parameters under polytomous IRT models requires some explanation. First, the "one-row-per-item" structure was preserved in this framework. Second, all models being different in terms of number of parameters per item, the number of columns in this bank will vary from one item bank and one model to another. A complete description therefore requires a detailed presentation of the polytomous IRT models.

2.1. Parametrization of polytomous models

For a given item j, let the response categories be coded as 0, 1, ..., g_j so that $g_j + 1$ response categories are available. Let X_j be the item response and θ the ability level of the respondent. Set also $P_{jk}(\theta) = \mathsf{P}(X_j = k|\theta)$ as the response category probability, that is, the probability that response category k ($k = 0, 1, \ldots, g_j$) is picked up for item j.

The GRM and MGRM belong to the class of so-called *difference models* (Thissen and Steinberg 1986) and are defined by means of *cumulative response probabilities* $P^*_{jk}(\theta) = \mathsf{P}(X_j \geq k|\theta)$, that is, the probability of selecting a response category in $\{k, k+1, \ldots, g_j\}$, and with the convention that $P^*_{j0}(\theta) = 1$ and $P^*_{jk}(\theta) = 0$ for any $k > g_j$. Response category probabilities are then computed as $P_{jk}(\theta) = P^*_{jk}(\theta) - P^*_{j,k+1}(\theta)$.

Using the notations given in Embretson and Reise (2000), the cumulative probability of the GRM takes the following form:

$$P^*_{jk}(\theta) = \frac{\exp\left[\alpha_j\left(\theta - \beta_{jk}\right)\right]}{1 + \exp\left[\alpha_j\left(\theta - \beta_{jk}\right)\right]}, \tag{1}$$

while the cumulative probability of the MGRM is written as:

$$P^*_{jk}(\theta) = \frac{\exp\left[\alpha_j\left(\theta - b_j + c_k\right)\right]}{1 + \exp\left[\alpha_j\left(\theta - b_j + c_k\right)\right]}. \tag{2}$$

The GRM allows thus for category threshold parameters β_{jk} that vary across items, while the MGRM assumes the same number of response categories for all items (i.e., $g_j = g$ for all items) and identical threshold parameters c_k across items.

The PCM, GPCM, RSM and NRM, on the other hand, belong to the class of *divide-by-total models* (Thissen and Steinberg 1986). The respective response category probabilities are set as follows:

$$P^*_{jk}(\theta) = \frac{\exp\sum_{t=0}^{k}\alpha_j\left(\theta - \delta_{jt}\right)}{\sum_{r=0}^{g_j}\exp\sum_{t=0}^{r}\alpha_j\left(\theta - \delta_{jt}\right)} \quad \text{with} \quad \sum_{t=0}^{0}\alpha_j\left(\theta - \delta_{jt}\right) = 0 \tag{3}$$

for the GPCM,

$$P^*_{jk}(\theta) = \frac{\exp\sum_{t=0}^{k}[\theta - (\lambda_j + \delta_t)]}{\sum_{r=0}^{g_j}\exp\sum_{t=0}^{r}[\theta - (\lambda_j + \delta_t)]} \quad \text{with} \quad \sum_{t=0}^{0}[\theta - (\lambda_j + \delta_t)] = 0 \tag{4}$$

for the RSM, and

$$P^*_{jk}(\theta) = \frac{\exp(\alpha_{jk}\,\theta + c_{jk})}{\sum_{r=0}^{g_j}\exp(\alpha_{jr}\,\theta + c_{jr})} \quad \text{with} \quad \alpha_{j0}\,\theta + c_{j0} = 0 \tag{5}$$

for the NRM. The PCM is a particular case of the GPCM (3) with the restriction $\alpha_j = 1$. The RSM assumes all items have an equal number of response categories (i.e., $g_j = g$ for all j), while other models allow for different numbers of response categories across items.

2.2. Specification of the item bank

In order to correctly specify the polytomous item bank in **catR**, it is first mandatory that the items be calibrated using the same parametrization of the models (1) to (5) above. Then, since each item will be coded as one row of the item bank, the ordering of the item parameters is central. It was decided to make use of the following ordering for any item j:

- for the GRM: $(\alpha_j, \beta_{j1}, \ldots, \beta_{j,g_j})$
- for the MGRM: $(\alpha_j, b_j, c_1, \ldots, c_g)$
- for the PCM: $(\delta_{j1}, \ldots, \delta_{j,g_j)})$
- for the GPCM: $(\alpha_j, \delta_{j1}, \ldots, \delta_{j,g_j)})$
- for the RSM: $(\lambda_j, \delta_1, \ldots, \delta_g)$
- for the NRM: $(\alpha_{j1}, c_{j1}, \alpha_{j2}, c_{j2}, \ldots, \alpha_{j,g_j}, c_{j,g_j})$

In other words, the number of columns in the item bank will vary from one model to another. If $g_{\max}$ stands for the maximum number of response categories across all items ($g_{\max} = g$ in case of MGRM and RSM), then the number of columns in the item bank (without the possible subgroup membership indicators) is $g_{\max} + 1$ for the GRM, MGRM, GPCM and RSM, $g_{\max}$ for the PCM and $2 \times g_{\max}$ for the NRM. If an item has less than the maximal number of response categories, the corresponding row of the item bank is completed by `NA` values for the missing response categories.

3. Additional item selection rules

The former version of the package included seven item selection rules, listed in Magis and Raîche (2012, p. 9). Now, **catR** holds five additional item selection rules that are briefly described below.

1. The *thOpt* procedure (Li and Schafer 2005; Magis 2013). In the *thOpt* rule, the item selected is the one belonging to the subset of administrable items of the bank (B) with minimum distance between the currently estimated trait level $\hat{\theta}$ and the value where the item achieves its maximum in the Fisher information function $\theta_i^{\max}$:

$$j = \arg\min_{i \in B} \left| \hat{\theta} - \theta_i^{\max} \right|. \tag{6}$$

   The computation of $\theta_i^{\max}$ is done with the equations provided in Magis (2013).

2. The *Kullback-Leibler divergency* criterion weighted by the *posterior distribution* (KLP; Chang and Ying 1996). The Kullback-Leibler (KL) information function evaluates the item discrimination capacity between any possible pairs of trait levels. This means that KL is a global information measure. Chang and Ying (1996) proposed to weight the KL measure with the posterior trait level distribution:

$$j = \arg\min_{i \in B} \int_{-\infty}^{+\infty} KL_i(\theta \| \hat{\theta})\, f(\theta)\, L(\theta)\, d\theta, \tag{7}$$

where $f(\theta)$ is the prior distribution of ability, $L(\theta)$ is the likelihood function and $KL_i(\theta\|\hat{\theta})$ is calculated as follows (see also van der Linden and Pashley 2010):

$$KL_i(\theta\|\hat{\theta}) = \mathsf{E}\left[\log \frac{L(\hat{\theta}|X_i)}{L(\theta|X_i)}\right] = \sum_{k=0}^{g_i} P_{ik}(\hat{\theta}) \log\left[\frac{P_{ik}(\hat{\theta})}{P_{ik}(\theta)}\right], \tag{8}$$

with $L(\theta|X_i)$ being the contribution term of item i to the full likelihood $L(\theta)$.

3. The *Kullback-Leibler divergency* criterion weighted by the *likelihood function* (KL; Barrada, Olea, Ponsoda, and Abad 2009b). In this version of the KL selection rule, no prior distribution is considered, so the item selected is:

$$j = \arg\min_{i \in B} \int_{-\infty}^{+\infty} KL_i(\theta\|\hat{\theta})\, L(\theta)\, d\theta. \tag{9}$$

4. The *progressive method* (Revuelta and Ponsoda 1998). In the progressive method the selected item is the one for which the weighted sum of a random component and the Fisher information is highest. At the beginning of the test, when the trait estimation error is high, the weight of the random component is maximum and the weight of the Fisher information is minimum. As the number of administered items increases (in fixed length CATs) or when the estimated standard error approaches the standard error threshold (when the `"precision"` rule is applied), the weight of the random component decreases and the weight of the Fisher information increases. The progressive method can be described as follows:

$$j = \arg\max_{i \in B} \lfloor (1 - W)\, R_i + W\, I_i(\hat{\theta}) \rfloor, \tag{10}$$

where R_i is a random number belonging to the interval $\left[0, \max_{i \in B} I_i(\hat{\theta})\right]$ and $I_i(\hat{\theta})$ is the Fisher information function computed at the $\hat{\theta}$ value.

For fixed length CATs, Barrada, Olea, Ponsoda, and Abad (2008) proposed the following equation to relate W to the number of item positions in the test (ranging from 1 to Q):

$$W = \begin{cases} 0 & \text{if } q = 1, \\ \frac{\sum_{f=1}^{q} (f-1)^t}{\sum_{f=1}^{Q} (f-1)^t} & \text{if } q \neq 1. \end{cases} \tag{11}$$

The t parameter marks the speed at which the weight of the random component is reduced, and thus the speed at which the importance of item information increases. Higher values imply a higher relevance of the random component in the item selection. When the stopping rule surpasses a predefined standard error value, the W value is computed with an adaptation of the method proposed by McClarty, Sperling, and Dodd (2006):

$$W = \max\left[\frac{I(\hat{\theta})}{I_{\text{stop}}}, \frac{q}{M-1}\right]^t, \tag{12}$$

where I_{stop} is the Fisher information required for reaching the standard error threshold and M is the maximum test length.

5. The *proportional method* (Barrada *et al.* 2008; Segall 2004). While the rest of the selection methods implemented in **catR** are deterministic, the proportional method is stochastic. The probability of selecting the item is given by:

$$\mathsf{P}(S_i) = \frac{z_i \, I_i(\hat{\theta})^H}{\sum_{k=1}^{n} z_k \, I_k(\hat{\theta})^H}, \tag{13}$$

where n is the size of the item bank and z_i indicates whether the item belongs (1) or not (0) to B. Once the probabilities of each item being selected are computed, a cumulative distribution of probabilities is derived. Then, a random number drawn from the uniform interval (0,1) is used to identify the item to be selected.

For fixed length CATs, Barrada *et al.* (2008) have proposed defining H as follows:

$$H = \begin{cases} 0 & \text{if } \; q = 1, \\ \dfrac{Q \sum_{f=1}^{q} (f-1)^s}{\sum_{f=1}^{Q} (f-1)^s} & \text{if } \; q \neq 1. \end{cases} \tag{14}$$

The s parameter has the same role as the t parameter in the progressive rule.

For the `"precision"` stopping rule, the computation of H is:

$$H = I_{\text{stop}} \, \max \left[\frac{I(\hat{\theta})}{I_{\text{stop}}}, \frac{q}{M-1} \right]^s. \tag{15}$$

Note that for clarity the formerly called *Urry*'s method (Urry 1970) has been renamed as the *bOpt* criterion. Moreover, all item selection rules are available for both dichotomous and polytomous IRT models, except the *thOpt* and the *bOpt* methods (which are restricted to dichotomous models). Also, the progressive and proportional methods are not available for classification CATs.

4. Post-hoc simulations

The generation of a CAT response pattern is done by random draws from the Bernoulli distribution for each item response. More precisely, once the next item to administer is selected, the probability of answering this item correctly, say $P_j(\theta)$, is computed with the estimate of ability θ and the item response X_j is drawn from the Bernoulli distribution with success probability $P_j(\theta)$. Note that by including polytomous IRT models, this random sampling scheme was updated by considering draws from the appropriate multinomial distribution.

The package **catR** now allows for *post-hoc simulations*, that is, item responses that are not randomly drawn but picked from a given response pattern. This response pattern is directly provided in the `randomCAT()` or `simulateRespondents()` functions with the newly added arguments `responses` and `responsesMatrix`, respectively (see Appendix A). By default, these arguments take the `NULL` value, so that item responses are randomly drawn from the appropriate (Bernoulli or multinomial) distribution. Otherwise, `responses` must be a vector and `responsesMatrix` a matrix of item responses (either dichotomous or polytomous) of the same length of the number of items in the bank, and with the same ordering (i.e., first item response to the first item in the bank etc.).

In the case of post-hoc simulations, the true ability level may not be provided (as it will be unknown in practical cases). The `randomCAT()` function nevertheless returns its value through the `trueTheta` argument, with a by-default value of zero (for compatibility with traditional random CAT generation).

In post-hoc simulations, when real examinees (not simulees) have responded to the full item bank, it is common to treat the estimated ability with the full vector of responses as the best guess of the true ability level. In those cases, the best `trueTheta` estimate could be obtained with the `thetaEst` function. Otherwise, `trueTheta` in the context of post-hoc simulations could be fixed to any arbitrary value. In any case, the `trueTheta` argument is not used for the generation of item responses.

The post-hoc simulation feature can be applied to at least two different situations. First, the responses of examinees to the full item bank are available and the user wants to evaluate the effects of switching from a linear test to an adaptive test (see, e.g., Fischer *et al.* 2014; Gibbons *et al.* 2008). Second, the responses to the items come from a previous phase of the simulation process and must remain constant in the adaptive phase. For instance, with post-hoc simulations it is possible to simulate the effects of item parameter calibration error in adaptive testing (Olea, Barrada, Abad, Ponsoda, and Cuevas 2012; van der Linden and Glas 2000). An example is also provided in the next section.

5. Illustration

Let us now illustrate briefly the main updates of **catR** by displaying the full code to generate CAT patterns. The main steps have been described in Magis and Raîche (2012) and will not be detailed here, emphasis being put on new topics instead.

Throughout this section the following options will be selected and kept identical across examples for sake of clarity (they can obviously be modified according to the user's interests).

1. An item bank of 500 items is randomly generated with the PCM as the IRT model. Moreover, each item has between two and five response categories.

2. Each CAT starts by selecting from the item bank, the item that is most informative for the true ability level of zero. This means, among others, that each CAT will start with the same item (this restriction can nevertheless be relaxed by using another approach; the current one, however, is commonly used in real CAT assessments).

3. Ad-interim ability is estimated with the maximum a posteriori (or Bayes modal) method, with the standard normal prior distribution of ability.

4. The next item to administer is selected by making use of the Kullback-Leibler (KL) divergency criterion.

5. The stopping rule is set as a precision criterion: Adaptive administration ends when the standard error of the ad-interim ability estimate becomes smaller than 0.3.

6. The final ability estimator is the traditional maximum likelihood (ML) estimator.

7. The examples do not contain any option for content balancing nor for item exposure control.

These baseline options can be implemented in R with the following code (see Magis and Raîche 2012, for further details):

```
R> library("catR")
R> bank <- genPolyMatrix(items = 500, model = "PCM", nrCat = 5)
R> start <- list(nrItems = 1, theta = 0)
R> test <- list(itemSelect = "KL", method = "BM")
R> stop <- list(rule = "precision", thr = 0.3)
R> final <- list(method = "ML")
```

Note that the item bank stored in `bank` is generated through the new function `genPolyMatrix()` that is further described in Appendix A.

The first rows of the generated item bank (stored into the R object `bank`) can be looked at for information:

```
R> head(bank)
```

leading to:

```
  deltaj1 deltaj2 deltaj3 deltaj4
1   0.136   0.407  -0.070      NA
2  -0.248   0.696   1.146      NA
3  -2.403   0.573   0.375  -0.425
4   0.951  -0.389  -0.284   0.857
5   1.720   0.270      NA      NA
6  -0.422  -1.189  -0.331  -0.940
```

According to the PCM parametrization in (3), items 1 and 2 hold four responses categories, items 3, 4 and 6 have five response categories and item 5 has only three response categories.

5.1. Example 1

In the first example, a single CAT pattern will be generated from the usual random response generation process, with a true ability level of one and all aforementioned CAT options. The corresponding R code is given below:

```
R> res <- randomCAT(trueTheta = 1, itemBank = bank, model = "PCM",
+    start = start, test = test, stop = stop, final = final)
```

The corresponding output is returned as follows:

```
Random generation of a CAT response pattern
  with random seed equal to 1

 Item bank calibrated under Partial Credit Model

 True ability level: 1
```

```
 Starting parameters:
   Number of early items: 1
   Early item selection: maximum informative item for starting ability
   Starting ability: 0

 Adaptive test parameters:
   Next item selection method: Kullback-Leibler (KL) information
   Provisional ability estimator: Bayes modal (MAP) estimator
     Provisional prior ability distribution: N(0,1) prior
   Ability estimation adjustment for constant pattern: none

 Stopping rule:
   Stopping criterion: precision of ability estimate
    Maximum SE value: 0.3

 Randomesque method:
   Number of 'randomesque' starting items: 1
   Number of 'randomesque' test items: 1

 Content balancing control:
   No control for content balancing

 Adaptive test details:

Nr        1     2     3     4     5     6
Item    439   363   321   200   492   100
Resp.     4     3     3     3     1     3
Est.   0.49 1.015 1.195 1.355  1.24   1.3
SE    0.668 0.488 0.387 0.352 0.308 0.281

 Satisfied stopping rule:
   Precision of ability estimate

 Final results:
   Length of adaptive test: 6 items
   Final ability estimator: Maximum likelihood estimator
   Final range of ability values: [-4,4]
   Final ability estimate (SE): 1.415 (0.305)
   95% confidence interval: [0.817,2.013]

 Output was not captured!
```

The CAT required only six item responses to reach the pre-specified level of precision in the ability estimation process. Note that the final SE value (0.305) is larger than the requested threshold (0.3), which is due to the change in ability estimator between the test and final steps. Moreover, the final ability estimate equals 1.415, not far from the true underlying ability level of one.

5.2. Example 2

In the second example, an illustration of post-hoc simulation is performed. First, for the sake of such analysis, some response patterns must be provided. Here we make use of the new `genPattern()` function to create this pattern (see Appendix A for further details), though in practical situations it is often provided from real test assessments.

```
R> x <- genPattern(th = 1, it = bank, model = "PCM")
```

Then, the CAT pattern is obtained using the following code. Note that in this context of post-hoc simulation, the true ability level may not be provided anymore (as it is only used to generate the item responses).

```
R> res2 <- randomCAT(itemBank = bank, responses = x, model = "PCM",
+    start = start, test = test, stop = stop, final = final)
R> res2
```

The output is very similar to the one for `res1` in the previous section, so only the specific parts are displayed below

```
Post-hoc simulation of a full bank provided response pattern

 [SKIPPED OUTPUT]

 Adaptive test details:

Nr        1     2     3     4     5     6
Item    439   363   200   492    71   321
Resp.     4     4     2     4     0     0
Est.   0.49 1.272 1.306 1.577 1.401 1.161
SE    0.668 0.535 0.421 0.395 0.335 0.288

[SKIPPED OUTPUT]
```

Here also only six items are required to fulfill the CAT stopping rule. In this case, it is worth checking that the item responses returned by the CAT process

```
R> res2$pattern
```

are actually equal to the item responses from the input response pattern

```
R> x[res2$testItems]
```

both returning (in this example) the same response pattern $(4, 4, 2, 4, 0, 0)$.

5.3. Example 3

In this final example, the new function `simulateRespondents()`, described in Appendix A, will be illustrated. An artificial set of 20 respondents is considered, with true ability levels being equally spaced between -2 and 2. All other CAT options remain unchanged. The full R code is displayed below.

```
R> thetas <- seq(from = -2, to = 2, length = 20)
R> res3 <- simulateRespondents(thetas = thetas, itemBank = bank,
+    model = "PCM", start = start, test = test, stop = stop, final = final)
R> res3
```

The output of this function is displayed in a somewhat different setting than the output from `randomCAT()`. That is, summary statistics on the whole set of simulated patterns are returned instead (though all individual results can be retrieved from the elements of the output list `res3`, for instance by calling `str(res3)`). This output is reproduced below.

```
 ** Simulation of multiple examinees **

Random seed was fixed (see argument 'genSeed')

Simulation time: 3.7835 minutes

Number of simulees: 20
Item bank size: 500 items
IRT model: PCM

Item selection criterion: KL
 Stopping rule:
         Stopping criterion: precision of ability estimate
           Maximum SE value: 0.3
 rmax: 1

Mean test length: 7.1 items
Correlation(true thetas,estimated thetas): 0.9222
RMSE: 0.5081
Bias: 0.0728
Proportion of simulees that satisfy the stop criterion: 1

Maximum exposure rate: 1
Number of item(s) with maximum exposure rate: 1
Minimum exposure rate: 0
Number of item(s) with minimum exposure rate: 445
Item overlap rate: 0.3141

Conditional results
                       Measure     D1     D2     D3     D4     D5     D6    D7
                    Mean Theta -1.895 -1.474 -1.053 -0.632 -0.211  0.211 0.632
                          RMSE  0.026  0.286  0.671  0.833  0.358   0.34 0.433
                     Mean bias  0.025  0.235 -0.557  0.647  0.342 -0.203 0.139
               Mean test length      9    5.5      7      5      6    4.5   5.5
             Mean standard error  0.322  0.323  0.327  0.287   0.28  0.292 0.286
Proportion stop rule satisfied      1      1      1      1      1      1     1
             Number of simulees      2      2      2      2      2      2     2
```

```
    D8    D9   D10
 1.053 1.474 1.895
 0.182  0.47 0.818
 0.124 -0.43 0.406
   5.5     5    18
   0.3 0.312 0.341
     1     1     1
     2     2     2

These results can be saved by setting 'save.output' to TRUE
  in the 'simulateRespondents' function
```

Note that the long computational time (about four minutes) is due to the use of the *KL* rule as method for next item selection, which is a very computationally intensive one. Other methods such as *MFI* for instance would reduce this computational effort to a few seconds instead.

6. Final comments

The R package **catR** offers a flexible routine to generate response patterns under a CAT scenario. It has many options for ability estimation, next item selection, item exposure and content balancing control, as well as several rules for selecting the first items and stopping the CAT. Both dichotomous and polytomous IRT models are now supported by **catR**, and post-hoc simulations can also be considered as an alternative to usual random response draws.

Practical applications of **catR** are numerous. First, it was originally developed as a research tool to perform intensive and comparative simulation studies. Up to now, a common dynamic in the research area of CAT has been that each researcher has developed his/her own code to perform the simulations. However, making use of a common package like **catR** would alleviate some related problems: (a) it reduces the time to implement CAT routines; (b) it provides more consistency in research and allows replication studies; and (c) it facilitates the use of more complex IRT models that are available in **catR**. Moreover, the modularity of its architecture and its open-source access implies that any researcher can use it, as it stands or by modifying some functions. The inclusion of polytomous IRT models and additional item selection rules will allow studies to broaden this area of research, for instance by comparing several items selection rules or ability estimators with various models and test situations.

The package **catR** can also be useful with real or simulated data. We can foresee several scenarios for which a free accessible alternative as **catR** can reduce costs.

1. Pre-operational analysis, to simulate the adequate protocol (item selection rule or trait level estimator) when considering to start a CAT implementation with real item banks.

2. Empirical evaluation of the gain in switching from linear to adaptive administration of previously developed and calibrated items using post-hoc simulations (e.g., Fischer *et al.* 2014; Gibbons *et al.* 2008).

3. Operational purposes, as the support for the platform of CAT administration. One example is the web-based platform **Concerto** (Kosinski *et al.* 2013) that requires **catR** as underlying computational routine for CAT administrations.

Note that **catR** is not the only R package devoted to adaptive testing. Among others, **mirtCAT** (Chalmers 2016) seems to be a valuable alternative. Its main asset is to allow the creation of graphical user interfaces for administering CATs in real time. **catR**, on the other hand, is more complete in terms of CAT options for selecting the first item(s), next item selection and stopping rules. In addition, **mirtCAT** package supports several multidimensional IRT models, which is currently not the case with **catR**.

Future updates of **catR** will focus on several modern aspects of CAT assessment. Some possible future extensions are: the inclusion of multidimensional IRT models (Reckase 2009); cognitive diagnosis CAT models (Cheng 2009; Kaplan, de la Torre, and Barrada 2015); new or other methods for item exposure and content balancing control; and testlet IRT models (Wainer, Bradlow, and Wang 2009).

Acknowledgments

David Magis is Research Associate of the *Fonds de la Recherche Scientifique – FNRS*, University of Liège, Belgium. Juan Ramon Barrada is Senior Lecturer, Universidad Zaragoza, Teruel, Spain. This research was supported by the IAP Research Network P7/06 of the Belgian State (Belgian Science Policy), and a grant from the *Fundación Universitaria Antonio Gargallo* and the *Obra Social de Ibercaja* (Spain). Correspondence should be sent to David Magis.

References

Andrich D (1978a). "Application of a Psychometric Model to Ordered Categories Which Are Scored with Successive Integers." *Applied Psychological Measurement*, **2**(4), 581–594. doi:10.1177/014662167800200413.

Andrich D (1978b). "A Rating Formulation for Ordered Response Categories." *Psychometrika*, **43**(4), 561–573. doi:10.1007/bf02293814.

Barrada JR, Abad FJ, Veldkamp BP (2009a). "Comparison of Methods for Controlling Maximum Exposure Rates in Computerized Adaptive Testing." *Psicothema*, **21**(2), 313–320.

Barrada JR, Olea J, Ponsoda V, Abad FJ (2008). "Incorporating Randomness to the Fisher Information for Improving Item Exposure Control in CATS." *British Journal of Mathematical and Statistical Psychology*, **61**(2), 493–513. doi:10.1348/000711007x230937.

Barrada JR, Olea J, Ponsoda V, Abad FJ (2009b). "Item Selection Rules in Computerized Adaptive Testing: Accuracy and Security." *Methodology*, **5**(1), 7–17. doi:10.1027/1614-2241.5.1.7.

Bock RD (1972). "Estimating Item Parameters and Latent Ability When Responses Are Scored in Two or More Nominal Categories." *Psychometrika*, **37**(1), 29–51. doi:10.1007/bf02291411.

Chalmers RP (2016). "Generating Adaptive and Non-Adaptive Test Interfaces for Multidimensional Item Response Theory Applications." *Journal of Statistical Software*, **71**(5), 1–39. doi:10.18637/jss.v071.i05.

Chang HH, Ying Z (1996). "A Global Information Approach to Computerized Adaptive Testing." *Applied Psychological Measurement*, **20**(3), 213–229. doi:10.1177/014662169602000303.

Cheng Y (2009). "When Cognitive Diagnosis Meets Computerized Adaptive Testing: CD-CAT." *Psychometrika*, **74**(4), 619–632. doi:10.1007/s11336-009-9123-2.

Embretson SE, Reise SP (2000). *Item Response Theory for Psychologists*. Lawrence Erlbaum Associates, Mahwah.

Fischer HF, Klug C, Roeper K, Blozik E, Edelmann F, Eisele M, Stor S, Wachter R, Scherer M, Rose M, Herrmann-Lingen C (2014). "Screening for Mental Disorders in Heart Failure Patients Using Computer-Adaptive Tests." *Quality of Life Research*, **23**(5), 1609–1618. doi:10.1007/s11136-013-0599-y.

Gibbons RD, Weiss DJ, Kupfer DJ, Frank E, Fagiolini A, Grochocinsk VJ, Bhaumik DK, Stover A, Bock RD, Immekus JC (2008). "Using Computerized Adaptive Testing to Reduce the Burden of Mental Health Assessment." *Psychiatric Services*, **59**(4), 361–368. doi:10.1176/appi.ps.59.4.361.

Kaplan M, de la Torre J, Barrada JR (2015). "New Item Selection Methods for Cognitive Diagnosis Computerized Adaptive Testing." *Applied Psychological Measurement*, **39**(3), 167–188. doi:10.1177/0146621614554650.

Kosinski M, Lis P, Mahalingam V, Kielczewski B, Okubo T, Stillwell D, Sun L, Rust J (2013). ***Concerto** – Open-Source Online R-Based Adaptive Testing Platfrom (Version 4)*. The Psychometrics Centre, Cambridge. URL http://www.psychometrics.cam.ac.uk/newconcerto.

Li YH, Schafer WD (2005). "Increasing the Homogeneity of CAT's Item-Exposure Rates by Minimizing or Maximizing Varied Target Functions While Assembling Shadow Tests." *Journal of Educational Measurement*, **42**(3), 245–269. doi:10.1111/j.1745-3984.2005.00013.x.

Maechler M, *et al.* (2016). ***sfsmisc**: Utilities from "Seminar für Statistik", ETH Zurich*. R package version 1.1-0, URL https://CRAN.R-project.org/package=sfsmisc.

Magis D (2013). "A Note on the Item Information Function of the Four-Parameter Logistic Model." *Applied Psychological Measurement*, **37**(4), 304–315. doi:10.1177/0146621613475471.

Magis D, Raîche G (2012). "Random Generation of Response Patterns under Computerized Adaptive Testing with the R Package **catR**." *Journal of Statistical Software*, **48**(8), 1–31. doi:10.18637/jss.v048.i08.

Masters GN (1982). "A Rasch Model for Partial Credit Scoring." *Psychometrika*, **47**(2), 149–174. doi:10.1007/bf02296272.

McClarty KL, Sperling RA, Dodd BG (2006). "A Variant of the Progressive-Restricted Item Exposure Control Procedure in Computerized Adaptive Testing Systems Based on the 3PL and Partial Credit Models." Paper presented at the Annual Meeting of the American Educational Research Association, San Francisco.

Muraki E (1990). "Fitting a Polytomous Item Response Model to Likert-Type Data." *Applied Psychological Measurement*, **14**(1), 59–71. doi:10.1177/014662169001400106.

Muraki E (1992). "A Generalized Partial Credit Model: Application of an EM Algorithm." *Applied Psychological Measurement*, **16**(2), 159–176. doi:10.1177/014662169201600206.

Olea J, Barrada JR, Abad FJ, Ponsoda V, Cuevas L (2012). "Computerized Adaptive Testing: The Capitalization on Chance Problem." *Spanish Journal of Psychology*, **15**(1), 424–441. doi:10.5209/rev_sjop.2012.v15.n1.37348.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Reckase M (2009). *Multidimensional Item Response Theory*. Springer-Verlag.

Revuelta J, Ponsoda V (1998). "A Comparison of Item Exposure Control Methods in Computerized Adaptive Testing." *Journal of Educational Measurement*, **35**(4), 311–327. doi:10.1111/j.1745-3984.1998.tb00541.x.

Samejima F (1969). "Estimation of Latent Ability Using a Response Pattern of Graded Scores." *Psychometrika Monograph*, **34**(Supplement 1), 1–97. doi:10.1007/bf03372160.

Samejima F (1996). "The Graded Response Model." In WJ van der Linden, RK Hambleton (eds.), *Handbook of Modern Item Response Theory*, pp. 85–100. Springer-Verlag.

Segall DO (2004). "A Sharing Item Response Theory Model for Computerized Adaptive Testing." *Journal of Educational and Behavioral Statistics*, **29**(4), 439–460. doi:10.3102/10769986029004439.

Thissen D, Steinberg L (1986). "A Taxonomy of Item Response Models." *Psychometrika*, **51**(4), 567–577. doi:10.1007/bf02295596.

Urry VW (1970). *A Monte Carlo Investigation of Logistic Test Models*. Unpublished doctoral dissertation, Purdue University, West Lafayette.

van der Linden WJ, Glas CAW (2000). "Capitalization on Item Calibration Error in Adaptive Testing." *Applied Measurement in Education*, **13**(1), 35–53. doi:10.1207/s15324818ame1301_2.

van der Linden WJ, Glas CAW (2010). *Elements of Adaptive Testing*. Springer-Verlag. doi:10.1007/978-0-387-85461-8.

van der Linden WJ, Pashley PJ (2010). "Item Selection and Ability Estimation in Adaptive Testing." In WJ van der Linden, CAW Glas (eds.), *Elements of Adaptive Testing*, pp. 3–30. Springer-Verlag. doi:10.1007/978-0-387-85461-8_1.

van der Linden WJ, Veldkamp BP (2004). "Constraining Item Exposure in Computerized Adaptive Testing with Shadow Tests." *Journal of Educational and Behavioral Statistics*, **29**(3), 273–291. doi:10.3102/10769986029003273.

Wainer H (2000). *Computerized Adaptive Testing: A Primer*. 2nd edition. Lawrence Erlbaum, Mahwah. doi:10.4324/9781410605931.

Wainer H, Bradlow ET, Wang X (2009). *Testlet Response Theory and Its Applications*. Cambridge University Press, Cambridge.

A. Additional updates and modifications

Together with previously described updates of the package, several technical modifications and improvements were performed. They are briefly listed below for completeness.

A.1. What remains unchanged

The general architecture of **catR** is such that some elements can be modified, updated or removed without needing to rewrite the whole package. Therefore, despite important improvements, the general structure of the package was left unchanged. That is, to generate a response pattern, one must provide a calibrated item bank in an appropriate format, a true ability level (or a full response pattern for post-hoc simulations), and four lists of options for the starting, testing, stopping and final steps of a CAT (see Figure 1 of Magis and Raîche 2012, p. 7 for further details). Hence, previous code developed for **catR** version 2.6 or before will remain valid with the most recent version of the package.

A.2. Removed or replaced features

The main modification in **catR** is the removal of the `createItemBank()` function and its replacement with a simpler function called `breakBank()`. The purpose of `createItemBank()` was to produce an item information matrix to quickly pick-up Fisher information for a given item and ability level. This structure was however not very user-friendly and required the creation and storage of an information matrix, and on-the-fly computation of information functions is very fast and straightforward with modern computers.

Another feature of `createItemBank()`, however, was to break down the item bank into two pieces (whenever supplied): the item parameters on the one hand and the subgroup membership of the items on the other hand (for content balancing purposes). This feature had to be preserved for proper functioning of **catR**, and this was achieved by creating the simpler function `breakBank()` instead. This new function takes as input the original matrix with both item parameters and subgroup membership and returns as output a list with the two elements. Note that `breakBank()` is used internally in the main function `randomCAT()` of **catR**, so now only the original, full matrix of item parameters (plus perhaps subgroup membership) must be supplied as input information in `randomCAT()`.

Note also that, in order to remove the former dependency of **catR** to the package **sfsmisc** (Maechler *et al.* 2016) for numerical integration, the updated package contains its own internal function for numerical integration, called `integrate.catR()`.

A.3. Item bank and response pattern generation

Two functions were created to automatically generate item banks according to a pre-selected IRT model. These functions are called `genDichoMatrix()` and `genPolyMatrix()` for dichotomous and polytomous IRT models, respectively. Both share four identical arguments: `items` to specify the requested number of items in the bank; `model` to determine the IRT model; `seed` to set the random seed; and `cbControl` to specify the options for content balancing control. In addition, `genDichoMatrix()` also allows specification of the parent distribution of each of the four parameters. `genPolyMatrix()`, on the other hand, requires the maximum number of item categories and can force the items to have exactly the same number of cat-

egories. The parent distributions, however, are currently set to default distributions. The interested reader can find more details about these functions in the **catR** help files.

Another useful function, called `genPattern()`, was created. As its name suggests, it performs random generation of a response pattern given a set of item parameters (argument `it`), a targeted ability level (argument `th`) and a pre-specified model, either dichotomous or polytomous IRT model (argument `model`). As already previously mentioned, this random generation is made by an appropriate call to the function `rbinom()` for dichotomous items and to `rmultinom()` for polytomous IRT models. The function returns a vector of random item responses with the same length of the number of rows in the item bank. Note that a single item can be specified by a vector of parameters (in the appropriate order according to the IRT model), and `genPattern()` converts it into an appropriate matrix for random response generation.

A.4. Multiple pattern generation

Finally, because the `randomCAT()` function can only produce one adaptive test at each call, an additional function was added to generate several response patterns simultaneously. This function, called `simulateRespondents()`, allows easy simulation of a large number CAT administrations and provides both statistical summaries and plots regarding accuracy and item exposure control. The results and plots are for the overall sample of examinees and is conditional on the deciles of the trait level distribution. Ten different plots can be displayed and saved. The availability of the plots depends on the stopping rule used. The details can be checked in the help files of the `simulateRespondents()` function.

The function `simulateRespondents()` makes use of most of the arguments of `randomCAT()`, with three main exceptions. First, the argument `trueTheta` is replaced by `thetas` and can hold a vector of true ability levels: Each value will be used to generate one response pattern with successive calls of `randomCAT()`. Second, in case of post-hoc simulations, the argument `responsesMatrix` contains a matrix of response patterns (one pattern per row) from which the item responses will be drawn. Third, two methods for controlling the maximum exposure rate that no item should surpass ($r^{\max}$) are available, the *restrictive* method (Revuelta and Ponsoda 1998) and the *item-eligibility* method (van der Linden and Veldkamp 2004). In both the restrictive and the item-eligibility methods, exposure control parameters k_i are used to define the subset B of the bank which is available for administration for each examinee and these parameters are computed on-the-fly, with each new examinee (Barrada, Abad, and Veldkamp 2009a).

In the restricted method, the control parameters can adopt just two values, 0 and 1. The k_i parameter will be set at 0 if the exposure rate of the item from the first CAT administration until the gth examinee $er_i^{(1\ldots g)}$ is greater than or equal to $r^{\max}$; otherwise, the control parameter will be set at 1:

$$k_i^{(g+1)} = \begin{cases} 1 & \text{if } er_i^{(1\ldots g)} < r^{\max}, \\ 0 & \text{if } er_i^{(1\ldots g)} \geq r^{\max}. \end{cases} \tag{16}$$

In the item-eligibility method, the parameters for the $(g+1)$th examinee are calculated considering $r^{\max}$, $er_i^{(1\ldots g)}$, and the exposure control parameters for the previous examinee $k_i^{(g)}$:

$$k_i^{(g+1)} = \begin{cases} 1 & \text{if } er_i^{(1\ldots g)}/k_i^{(g)} \leq r^{\max}, \\ \frac{r^{\max} k_i^{(g)}}{er_i^{(1\ldots g)}} & \text{if } er_i^{(1\ldots g)}/k_i^{(g)} > r^{\max}. \end{cases} \quad (17)$$

The k parameters determine the probability that each item is eligible for administration. For each item and with each new examinee, a random number from the interval $(0, 1)$ is generated and the item is marked as eligible only if that random number is lower than the k parameter.

Although the restrictive and item-eligibility methods do not exhaust all the possible methods for controlling the maximum exposure rate, those two options can be considered among the best available ones (Barrada *et al.* 2009a).

Affiliation:

David Magis
Department of Education (B32)
University of Liège
Boulevard du Rectorat 5
B-4000 Liège, Belgium
E-mail: david.magis@ulg.ac.be

Juan Ramon Barrada
Universidad de Zaragoza
Calle Ciudad Escolar, 2
44003 Teruel, Spain
E-mail: barrada@unizar.es

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
January 2017, Volume 76, Code Snippet 1 *Submitted:* 2014-12-01
doi:10.18637/jss.v076.c01 *Accepted:* 2015-11-02