Journal of Statistical Software

May 2012, Volume 48, Issue 8. *http://www.jstatsoft.org/*

Random Generation of Response Patterns under Computerized Adaptive Testing with the R Package catR

David Magis
University of Liège

Gilles Raîche
Université du Québec à Montréal

Abstract

This paper outlines a computerized adaptive testing (CAT) framework and presents an R package for the simulation of response patterns under CAT procedures. This package, called **catR**, requires a bank of items, previously calibrated according to the four-parameter logistic (4PL) model or any simpler logistic model. The package proposes several methods to select the early test items, several methods for next item selection, different estimators of ability (maximum likelihood, Bayes modal, expected a posteriori, weighted likelihood), and three stopping rules (based on the test length, the precision of ability estimates or the classification of the examinee). After a short description of the different steps of a CAT process, the commands and options of the catR package are presented and practically illustrated.

Keywords: computerized adaptive testing, item selection, ability estimation, R package.

1. Introduction

Computerized adaptive testing (CAT) was developed as an alternative to sequential (or fixed item) tests. In this CAT framework, the items of the test are assigned to each respondent in an optimal sequence, each item being selected as the most useful or informative at the current step of the test. Optimality is usually defined in regards to the previously administered items, the examinee's responses to these items, and the current or provisional ability estimate. Any CAT has several well-known advantages over sequential tests: it requires shorter tests to get the same level of precision in the ability estimates, it reduces the risk of fraud or cheating (each examinee having a different sequence of test items), it provides an immediate estimation of the ability level of the respondent (Wainer 2000). However, CAT has two main drawbacks. First, computer and software resources are necessary to administer adaptive tests, which might

become a practical issue when large groups of examinees have to be evaluated. Second, the optimal selection of the items can be achieved only if a sufficiently large and well calibrated bank of items is available. Both issues usually lead to expensive administrations of CAT (Wainer 2010). Despite its drawbacks, CAT has become an important field of development in the past three decades, as can be seen from the reference works by Drasgow and Olson-Buchanan (1999), Meijer and Nering (1999), Parshall, Spray, Kalohn, and Davey (2002), van der Linden and Glas (2000), Wainer (2000) and Weiss (1983).

Nevertheless, there is still a lack of available and freely accessible software to perform automatic generation of response patterns with a given item bank. The **CATSim** software (Assessment Systems Corporation 2012), previously known as **POSTSIM**, seems to be one of the most complete commercial software for CAT analysis and simulation. In the field of non-commercial software, the **Firestar** software (Choi 2009) and the `SIMCAT` SAS macro (Raiche and Blais 2006) are viable alternatives.

The purpose of this paper is twofold. First, the general principles of a CAT process are outlined, from the selection of the early first items to the final estimation of subject's ability. Second, we present a newly developed R package (R Development Core Team 2012), called **catR** (Magis and Raiche 2011), which is primarily designed for the generation of response patterns under CAT. The package integrates several rules for early item selection, ability estimation, next item selection, and stopping criteria. It is flexible with respect to the choice of the underlying logistic item response model and can handle large numbers of pattern generation for given true ability levels. In addition, it can control for crucial issues such as item exposure and content balancing.

The usefulness of **catR** is threefold. First, as an open-source package, it can be easily updated or modified by R-oriented users. Second, like any other R package, it can be used in interaction with other software or interface as underlying platform that performs all calculations (similarly to **Firestar** software, which has a user interface and an underlying R script for performing CAT computations). Third, the quality of item banks can be assessed by generating a large number of response patterns under various settings (different ability levels, different item selection rules, etc.) and comparing the results from simulated patterns, which can be easily performed with **catR**. This package is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=catR`.

The paper is organized as follows. In Section 2, we present the main IRT aspects that are involved in a CAT process: the underlying item response model and the methods of ability estimation. The different phases of an adaptive test are sketched in Section 3. They include the structure of an item bank, the selection of the first test items, the selection of the next item, the stopping rule, the final ability estimation, and the issues of item exposure and content balancing. Then, Section 4 displays the practical R code and related options for generating response patterns under a CAT process. This code is detailed for each step of the process as discussed in Section 3. Finally, some examples are proposed in Section 5 to illustrate the **catR** package.

2. IRT terminology

The underlying framework for developing computerized adaptive tests is the item response theory (IRT) framework. It permits the calibration of the item bank by specifying appropriate

item parameter values, to estimate the ability level of the subjects, and to select the next item(s) to be administered.

2.1. IRT models

Although the usual item response model under consideration for CAT is the three-parameter logistic (3PL) model (Birnbaum 1968), a less popular but more general model will be considered throughout this paper. The four-parameter logistic (4PL) model was introduced by Barton and Lord (1981) and takes the following form:

$$P_j(\theta) = \mathsf{P}(X_j = 1|\theta, a_j, b_j, c_j, d_j) = c_j + (d_j - c_j)\,\frac{\exp\left[a_j\,(\theta - b_j)\right]}{1 + \exp\left[a_j\,(\theta - b_j)\right]}. \tag{1}$$

In (1), X_j is the (binary) response of the examinee to item j; θ is the ability level of the examinee; and a_j, b_j, c_j and d_j are the four item parameters, respectively the discrimination parameter (the slope of the item characteristic curve), the difficulty level (the intercept), the pseudo-guessing parameter (the lower asymptote) and the inattention parameter (the upper asymptote). The 3PL and 4PL models differ only by the last parameter, the upper asymptote d_j, which is equal to one in the 3PL model but can be smaller than one in the 4PL model. In fact, $1 - d_j$ represents the probability that a high-ability examinee incorrectly answers the item.

The main reason for considering this 4PL model in a CAT environment is that it is a more general model than the usual logistic IRT models. As pointed out above, the 3PL model is a particular 4PL model where all d_j parameters equal one. If the lower asymptotes c_j are additionally constrained to the value zero, then only item difficulties and discriminations are present, and one deals with the two-parameter logistic (2PL) model. Finally, fixing all discriminations a_j's to one leads to the one-parameter logistic (1PL) model. The latter is also sometimes referred to as the Rasch model, although strictly speaking this model is obtained when all discriminations are fixed to one (unlike the 1PL model for which all a_j's are equal, but not necessarily to one). Also, the 4PL model has been introduced very recently in the CAT framework as an appropriate model to avoid issues with early mistakes in the test (Rulison and Loken 2009). This problem is particularly crucial for high-ability examinees that fail the first items of the adaptive test, since the recovery of the true ability level by its estimate cannot be guaranteed. Note that this is not an intrinsic property of CAT process but a weakness of IRT scoring (Green 2011). However, according to Rulison and Loken (2009), this issue can be withdrawn by using a 4PL model with upper asymptotes close to, but smaller than one.

The main issue with this 4PL model, however, is that the item parameters cannot be easily estimated. Like the estimation of the lower asymptote in the 3PL model, the estimation of the upper asymptote in the 4PL model requires a lot of subjects. Rulison and Loken (2009) avoided that issue by calibrating the item parameters under a 3PL model (for which convenient software routine exists) and artificially fixing the upper asymptotes to 0.99 or 0.98. Although the purpose of their study was not affected by this artefact, it is definitely not an appropriate method for item parameter estimation. Recently, Loken and Rulison (2010) proposed a Bayesian framework to estimate simultaneously the item parameters from the 4PL model. They found that the recovery of these four item parameters is quite good. Also, they showed the potential superiority of the 4PL model over the simplified 3PL, 2PL and 1PL models when the true one is the 4PL. In sum, the 4PL model has to be considered here

as a general model, from which usual logistic models can be found back by constraining the parameters appropriately, leaving the door open for future developments in estimation of the 4PL model.

Several methods are available for calibrating (i.e., estimating) the item parameters from a data set of response patterns. The most common methods are: joint maximum likelihood (Lord 1980), conditional maximum likelihood (Andersen 1970, 1972), marginal maximum likelihood (Bock and Aitkin 1981) and Bayesian modal estimation (Swaminathan and Gifford 1985, 1986). However, in the context of adaptive testing, it is often assumed that the item bank has been calibrated in advance, either from previous administrations of the items or by applying item generation methods (Irvine and Kyllonen 2002). The issue of estimating item parameters will therefore be skipped from this presentation, and it is assumed that one can provide a matrix of item parameters as a basis for constructing an item bank (see later).

It is also important to recall that the item responses are considered as binary, true-false responses. An obvious extension would consist in administrating polytomous, multiple choice items, or a mix of both types. Nevertheless, polytomous item responses can always be reduced to binary outcomes, this introduces some lack of information but allows for a complete application of the present CAT process. Furthermore, the estimation of ability levels is an entire part of the CAT framework and will consequently be discussed further.

2.2. Ability estimation

There exist several methods to estimate the ability level of an examinee, given the fixed item parameters and the corresponding response pattern. The most popular methods are the maximum likelihood (ML) estimator, the Bayes modal (BM) or maximum a posteriori estimator, the expected a posteriori (EAP) estimator and the weighted likelihood (WL) estimator. These methods are briefly presented below; see e.g., van der Linden and Glas (2000) for further details. All four estimators are available in the **catR** package.

The maximum likelihood estimator (Lord 1980) is the ability value $\hat{\theta}_{ML}$ that maximizes the likelihood function $L(\theta)$ or its logarithm $\log L(\theta)$:

$$L(\theta) = \prod_{j=1}^{J} P_j(\theta)^{X_j} Q_j(\theta)^{1-X_j} \text{ and } \log L(\theta) = \sum_{j=1}^{J} \{X_j \log P_j(\theta) + (1 - X_j) \log Q_j(\theta)\} \quad (2)$$

where $Q_j(\theta) = 1 - P_j(\theta)$ is the probability of an incorrect answer and J is the test length. The asymptotic standard error of the ML estimate can be approximated by the following formula:

$$se(\hat{\theta}_{ML}) = \frac{1}{\sqrt{\sum_{j=1}^{J} I_j(\hat{\theta}_{ML})}} \quad (3)$$

where $I_j(\theta)$ is the item information function:

$$I_j(\theta) = \frac{[P'_j(\theta)]^2}{P_j(\theta)\, Q_j(\theta)} \quad (4)$$

and $P'_j(\theta)$ stands for the first derivative of $P_j(\theta)$ with respect to θ.

The Bayes modal (BM) estimator (Birnbaum 1969) is similar to the ML estimator, except that the function to be maximized is the posterior distribution $g(\theta)$ of the ability level, which

is obtained by a combination of the prior distribution $f(\theta)$ and the likelihood function $L(\theta)$: $g(\theta) = f(\theta)L(\theta)$. In other words, the ML and BM estimators are the modes of the likelihood function and the posterior distribution, respectively. Thus, the BM estimator is the ability value $\hat{\theta}_{BM}$ that maximizes the posterior distribution $g(\theta)$ or its logarithm:

$$\log g(\theta) = \log f(\theta) + \log L(\theta) \tag{5}$$

The choice of a prior distribution is usually driven by some prior belief of the ability distribution among the population of examinees. The most common choice is the normal distribution with mean μ and variance σ^2. In this case, the standard error of $\hat{\theta}_{BM}$ is obtained by

$$se(\hat{\theta}_{BM}) = \frac{1}{\sqrt{\frac{1}{\sigma^2} + \sum_{j=1}^{J} I_j(\hat{\theta}_{BM})}}. \tag{6}$$

Sometimes, the prior mean and variance are fixed to zero and one, respectively, so that $f(\theta)$ reduces to the standard normal distribution. Another common choice is the uniform distribution on a fixed ability interval. In this case, both ML and BM estimators are equivalent when the ability interval of the prior uniform density is sufficiently wide.

Although less frequently considered in CAT, a third prior distribution can be considered: the Jeffreys' non informative prior density (Jeffreys 1939, 1946). This prior distribution is proportional to the square root of the test information function:

$$f(\theta) \propto \sqrt{I(\theta)} \tag{7}$$

and the test information function $I(\theta)$ is the sum of item information functions:

$$I(\theta) = \sum_{j=1}^{J} I_j(\theta) = \sum_{j=1}^{J} \frac{[P_j'(\theta)]^2}{P_j(\theta)\, Q_j(\theta)}. \tag{8}$$

With Jeffreys' prior distribution, the standard error of $\hat{\theta}_{BM}$ is approximated by

$$se(\hat{\theta}_{BM}) = \frac{1}{\sqrt{\frac{I'(\hat{\theta}_{BM})^2 - I(\hat{\theta}_{BM})\, I''(\hat{\theta}_{BM})}{2\, I(\hat{\theta}_{BM})^2} + \sum_{j=1}^{J} I_j(\hat{\theta}_{BM})}} \tag{9}$$

where $I'(\theta)$ and $I''(\theta)$ are respectively the first and second derivatives of $I(\theta)$ with respect to θ. Jeffreys' prior is said to be non-informative because it relies on the item parameters of the test, and not on some prior belief of the ability distribution, as modelled by the normal distribution for instance. It has the advantage of being less affected by misspecifications of the prior distribution. Its practical usefulness in the framework of CAT, however, still has to be validated.

The third estimator is the expected a posteriori (EAP) estimator (Bock and Mislevy 1982). While the BM estimator computes the mode of the posterior distribution, the EAP estimator computes its posterior mean:

$$\hat{\theta}_{EAP} = \frac{\int_{-\infty}^{+\infty} \theta\, f(\theta)\, L(\theta)\, d\theta}{\int_{-\infty}^{+\infty} f(\theta)\, L(\theta)\, d\theta} \tag{10}$$

with the same notations as for the BM estimator. In practice, the integrals in (10) are approximated, for instance by adaptive quadrature or numerical integration. The standard error of $\hat{\theta}_{EAP}$ is given by

$$se(\hat{\theta}_{EAP}) = \left[\frac{\int_{-\infty}^{+\infty} (\theta - \hat{\theta}_{EAP})^2 \, f(\theta) \, L(\theta) \, d\theta}{\int_{-\infty}^{+\infty} f(\theta) \, L(\theta) \, d\theta} \right]^{1/2} \tag{11}$$

Because the distribution of the ability levels is usually symmetric around the average ability level, the BM and the EAP estimators often return similar estimates and standard errors.

The last estimator to be presented here is the weighted likelihood (WL) estimator (Warm 1989). It was introduced to reduce and almost cancel the bias of the ML estimator, by using an appropriate weighing of the likelihood function. Although the ML estimator is asymptotically unbiased, Lord (1983, 1984) noticed that for small tests, its bias is proportional to the inverse of the test length. Warm (1989) established that the WL estimator $\hat{\theta}_{WL}$ must satisfy the following relationship:

$$\frac{J(\theta)}{2\, I(\theta)} + \frac{d \log L(\theta)}{d\, \theta} = 0 \tag{12}$$

where

$$J(\theta) = \sum_{j=1}^{J} \frac{P_j'(\theta) \, P_j''(\theta)}{P_j(\theta) \, Q_j(\theta)} \tag{13}$$

and $P_j''(\theta)$ is the second derivative of $P_j(\theta)$ with respect to θ (Warm 1989). Also, the standard error of $\hat{\theta}_{WL}$ is given by

$$se(\hat{\theta}_{WL}) = \frac{1}{\sqrt{\frac{I'(\hat{\theta}_{WL}) \, J(\hat{\theta}_{WL}) - I(\hat{\theta}_{WL}) \, J'(\hat{\theta}_{WL})}{2\, I(\hat{\theta}_{WL})^2} + \sum_{j=1}^{J} I_j(\hat{\theta}_{WL})}} \tag{14}$$

and $J'(\theta)$ is the first derivative of $J(\theta)$ with respect to θ.

In fact, $J(\theta)$ is the first derivative of the weight function with respect to θ, but the latter has no algebraic expression under the general 3PL model. Interestingly, Hoijtink and Boomsma (1995) and Warm (1989) noticed that under the 1PL and the 2PL models, the WL estimator is completely equivalent to the BM estimator with Jeffreys' prior distribution; see also Meijer and Nering (1999).

3. Principles of CAT

Any CAT process requires a calibrated item bank and can be split into four steps. The first step is the *initial step* and consists in selecting one or several appropriate items as the first test items. The second step is the *test step*, in which the items are successively chosen from the bank and the ability level is re-estimated after each item administration. The third step is the *stopping step* and sets the parameters for the stopping rule of the CAT. The *final step* yields the final estimation of ability level and possibly other additional information. Figure 1 is a schematic representation of the full CAT process, including the four steps. These are further presented in the next sections. Two additional CAT-related topics are also briefly outlined: item exposure and content balancing.

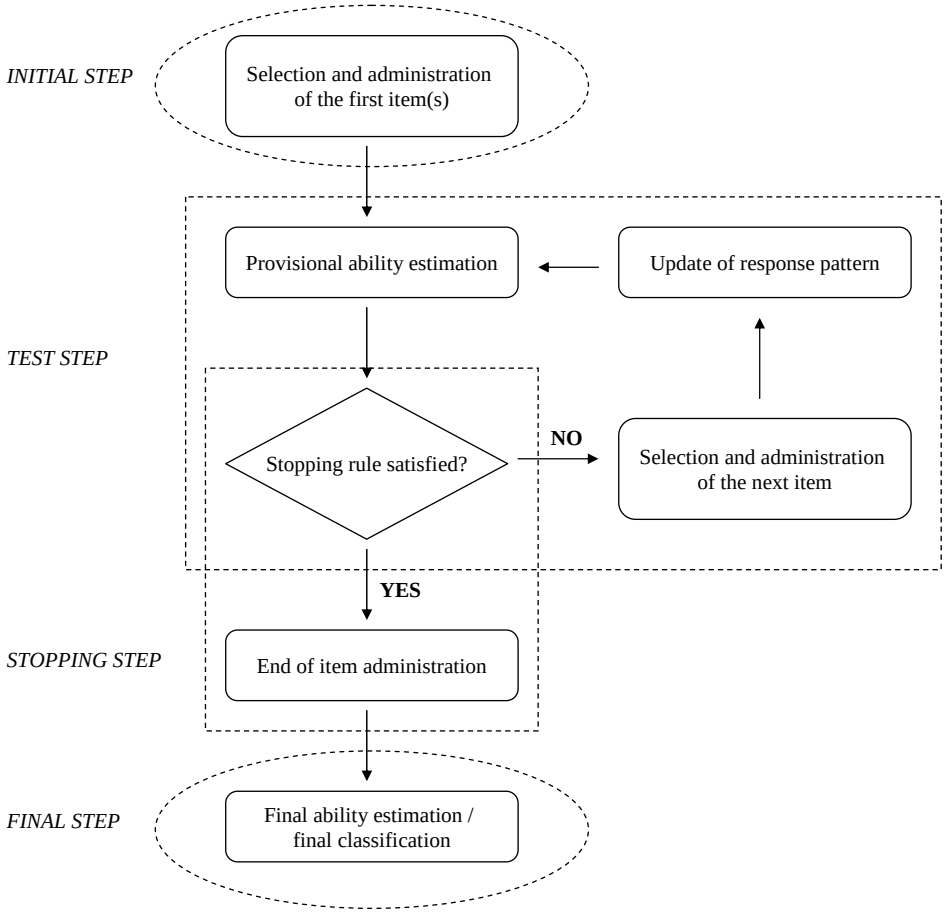

Figure 1: Schematic representation of a CAT process.

3.1. Item bank

The central tool for adaptive tests is the *item bank*. An item bank is a collection of items that can be administered to the examinees. In order to generate response patterns, it is sufficient to have access to the item parameter values. In this framework, the item bank is assumed to be calibrated prior to the start of the CAT process. That is, a large collection of items, which have been administered and calibrated by many anterior studies about the same topic, is available for adaptive testing. This might be a strong practical assumption with heavy financial impact, but this remains a realistic assumption anyway.

Very little is known on how large an item bank should be for optimal adaptive testing. The larger the item bank the better it is for CAT process, but it is not always possible to construct and to calibrate a large number of items. Also, a balanced item bank should contain items on the whole range of difficulty levels, from very easy to very difficult items. This would enable the accurate estimation of extreme ability levels. The absence of difficult items does not allow estimates of very large ability levels, while too many difficult items in the bank are not adequate for the estimation of low ability levels. Easy items are most informative for low ability levels, while difficult items are most informative for high ability levels.

3.2. Initial step

In order to start any CAT process, one has to select at least one item in the item bank and to administer it to the examinee. Most often, a single item is selected at this step, and in the absence of any prior information about the examinee's ability level, one fixes this ability level equal to the prior mean ability value, usually zero. With this prior belief, the initial item is selected as the most informative in the item bank for this ability value.

Although this is the standard approach, it can be improved by many aspects. First, if some prior information is available about the examinee, it can be incorporated into the initial step. For instance, knowing from previous tests that the examinee has rather high or low ability level, one can adjust the initial ability level to values larger or smaller than the average prior ability level, respectively. Another slight modification concerns the criterion for selecting the first item. As pointed out above, the most informative item is the standard choice, but one could consider as optimal selection, the item whose difficulty level is closest to the prior ability value. This reflects some intuitive reasoning that the most adequate item has a difficulty level very close to the ability level of the examinee. This is known as Urry's rule for selecting the next item (Urry 1970), but it is rarely applied at the initial step of a CAT.

A final improvement is to select more than one item at the initial step. This possibility is not clearly stated in the literature, and most of the CAT software permit to select only the first item in this initial step. However, another approach could be to select two or three items, each item referring to a different prior ability level in order to cover some range of abilities. For instance, fixing two prior ability levels to -1 and 1, and selecting the corresponding items (one per ability level), might reduce the issue of inadequate early item administration because of a lack of information about the subject's ability.

3.3. Test step

Once the initial items have been administered, one can get a first provisional estimate of ability by using the current set of responses. The second part of CAT, the test step, can be sketched as follows.

(a) Estimate the ability level by using the currently available information (previous response pattern to which is added the latest administered items). Set this as the provisional ability estimate.

(b) Select the next item among the bank items that have not been administered yet, according to the provisional ability estimate and the method for the next item selection.

(c) Administer the selected item to the examinee. Update the response pattern.

(d) Repeat steps (a) to (c) until the stopping criterion is satisfied (see later).

Any kind of ability estimator can be considered for the test step, but the ML estimator is often avoided because it often returns infinite estimates at the early steps of the adaptive test. The BM estimator with a prior normal distribution is a common choice for the test step.

In general, the same estimator is considered throughout the test step. However, it is possible to build a hybrid rule, starting with one ability estimator and switching to another one at some stage of the process. For instance, one could start with a Bayesian estimator, and switch

Criterion	Objective function	Optimization
MFI	$I_j(\hat{\theta}_{k-1}(\boldsymbol{X}))$	Maximize
MEPV	$\sum_{t=0}^{1} \mathsf{P}[X_j = t\|\hat{\theta}_{k-1}(\boldsymbol{X})]\, Var_j(\theta\|f(\theta), \boldsymbol{X}, t)$	Minimize
MLWI	$\int_{-\infty}^{+\infty} I_j(\theta)\, L(\theta\|\boldsymbol{X})\, d\theta$	Maximize
MPWI	$\int_{-\infty}^{+\infty} I_j(\theta)\, f(\theta)\, L(\theta\|\boldsymbol{X})\, d\theta$	Maximize
MEI	$\sum_{t=0}^{1} \mathsf{P}[X_j = t\|\hat{\theta}_{k-1}(\boldsymbol{X})]\, I_j(\hat{\theta}_k(\boldsymbol{X}, t))$	Maximize

Table 1: Criteria for next item selection and related objective functions. The "Optimization" column indicates whether the objective function must be maximized or minimized over all available items.

to the ML estimator when the response pattern has at least one success and one failure. The first estimator avoids infinite ability estimates by the inclusion of a prior density, while the ML estimator works independently of any prior distribution. Similarly, the WL estimator could be considered as it is less biased than the ML estimator. Other hybrid rules could be constructed similarly, by taking the advantages and drawbacks of each method into account.

There are several methods for next item selection. The most known ones are: maximum information criterion (MFI), minimum expected posterior variance (MEPV criterion), Urry's criterion (Urry 1970), maximum likelihood weighted information (MLWI) criterion (Veerkamp and Berger 1997), maximum posterior weighted information (MPWI) criterion (van der Linden 1998), maximum expected information (MEI) criterion (van der Linden 1998), and completely random selection. A brief overview of most of these techniques is proposed in Choi and Swartz (2009) and van der Linden and Pashley (2000).

In order to provide a detailed description of these criteria for next item selection, we set the following notations. Assume that $k-1$ items have been administered. Set $\boldsymbol{X}$ as the current response pattern, made by the $k-1$ responses to the first administered items, and set $\hat{\theta}_{k-1}(\boldsymbol{X})$ as the provisional ability estimate after the first $k-1$ items. If item j has not yet been administered, set $\hat{\theta}_k(\boldsymbol{X}, X_j)$ as the provisional ability estimate when this item j is administered and its response X_j is included into the current response pattern. We further refer to the items not yet administered as the *set of available items* and we denote this set by S_{k-1}. Note that the subscripts $k-1$ and k refer to the length of the current response pattern and not to any item number. Moreover, the notations $L(\theta|\boldsymbol{X})$ and $I_j(\theta|\boldsymbol{X})$ refer respectively to the likelihood function (2) and the item information function (4) evaluated at θ and given the response pattern $\boldsymbol{X}$. We also denote by $\mathsf{P}(X_j = t|\theta)$, the probability (1) that the response X_j to item j is equal to t, and t is either zero or one for incorrect and correct responses, respectively. Finally, $f(\theta)$ stands the prior ability distribution, and $Var_j(\theta|f(\theta), \boldsymbol{X}, t)$ is the posterior variance of θ, given its prior distribution, the current response pattern $\boldsymbol{X}$ augmented by item j whose response value X_j is equal to t.

Table 1 summarizes the objective functions of each of five of the criteria listed above. They correspond to the functions to be maximized or minimized in order to determine the best item to be administered next. The maximization (or minimization) is taken among all items in S_{k-1}, that is, among all available items. Applying either the maximization or minimization is listed in the "Optimization" column of Table 1.

With the MFI criterion, the next item is selected as the available item with maximum Fisher information at the current ability estimate. If these information functions were previously

computed into an information matrix, the selection of the next item by MFI is very fast. However, this relies on the provisional ability estimate, which might be severely biased, especially in the first steps of the test. The MLWI and MPWI criteria overcome this problem, from a marginalization process, by selecting the available item that maximizes a weighted form of the information function. For the MLWI criterion, the information is weighted by the likelihood function of the items already administered, while for the MPWI criterion, it is the posterior distribution of that acts as weighing function.

Instead of maximizing the information function, or some weighted form, over the current available items, another approach is to compute some expected optimal function based on the possible responses to the next item administered. Two criteria of that kind are the MEPV and the MEI criteria. In both cases, the optimal function is obtained by computing the probabilities of answering the next item correctly or incorrectly, and by updating the response pattern conditionally upon these two possible responses. With the MEI criterion, the optimal function is the expected information function, and with the MEPV, it is the expected posterior variance of the ability level. Both methods require the computation of the ability estimate, or its posterior variance, when the response pattern is updated either by a zero (incorrect response) or a one (correct response). The next item to be administered is the one among all available items that maximizes the expected information function (for the MEI criterion) or that minimizes the expected posterior variance (for the MEPV criterion).

Another method was proposed by van der Linden and Pashley (2000), the so-called Maximum Expected Posterior Weighted Information (MEPWI) criterion, as a combination of both MEI and MPWI criteria. However, Choi and Swartz (2009) demonstrated that this method is completely equivalent to the MPWI approach. For this reason, the MEPWI was not considered in this paper.

Finally, the last two possible methods are Urry's criterion and the completely random selection. Urry's criterion consists in selecting the available item whose difficulty level is as close as possible to the provisional ability estimate (Urry 1970). This is a straightforward method, and under the 1PL model it is completely equivalent to the MFI criterion. With other models however, some slight differences can occur. Completely random selection of the next item consists in a random draw from the set of available items. This might not be the optimal method yielding most informative tests, but its simplicity justifies its presence into the **catR** package. Moreover, performing some random item selection during the test might reduce the risk of item over exposure (Georgiadou, Triantafillou, and Economides 2007), at the risk of selecting less informative items. It mostly serves as a baseline method that can be compared against other, more efficient methods.

3.4. Stopping step

Any CAT process stops when the stopping criterion is fulfilled. Three main stopping rules are considered: the *length* criterion, the *precision* criterion and the *classification* criterion.

The *length* criterion imposes a maximum number of items to be administered. The CAT process stops when this maximal test length is attained. Longer tests obviously increase the precision of the estimates of ability, but shorter tests might be considered for investigating some issues in the early steps of a CAT, e.g., the effects of early mistakes on the item selection process (Rulison and Loken 2009).

The *precision* criterion forces the CAT process to stop as soon as the precision of the pro-

visional ability estimate reaches the pre-specified level of precision. Typically, the precision is measured by the standard error of the ability estimate, the lower the standard error the better the precision. Thus, one usually fixes a standard error as threshold value, and items are iteratively administered until the standard error of the provisional ability estimate gets smaller than or equal to that threshold.

Finally, the *classification* criterion is often used when the goal of the CAT is to classify examinees with respect to some ability threshold, rather than to estimate their ability level. For instance, one is interested in classifying students according to whether their ability level is larger or smaller than the value 0.5. Then, one examinee will be flagged with ability higher than 0.5 if there is enough confidence to assess this classification, according to the CAT response pattern. In practice, a confidence interval for the ability is built at each step of the test, and the CAT process goes on until the ability threshold is not included in the confidence interval anymore. This implies that the subjects' ability is either larger or smaller than the threshold, and the test stops. In sum, two parameters are needed to set up the classification criterion: the ability threshold and the confidence level of the interval. The larger the confidence, the longer the test will be to obtain a final classification. Also, very large or very small thresholds often lead to shorter tests, because it is always easy to discriminate examinees with extreme abilities (high or low) from middle-level ability examinees.

3.5. Final step

The final step of a CAT process provides the final estimation of the examinee's ability level, using the full response pattern to the adaptive test. The standard error of the estimate can also be displayed, and in the case of the classification stopping rule, the final classification of the examinee is also available. Any of the four estimators (ML, BM, EAP or WLE) can be considered in this final step, and there is no reason to use a different one than in the test step. However, it is possible to combine different estimators in the test and the final steps. For instance, a Bayesian estimator (EAP or BM) or the weighted likelihood method can be used throughout the test, and especially in the first steps of the process, to avoid the issue of infinite estimates with fully correct or fully incorrect patterns. At the final stage, however, the simple ML estimator can be used in order to get a final estimate that is free of any prior or weighing system.

3.6. Item exposure and content balancing

Apart from the four aforementioned steps, two additional issues are often controlled during a CAT process.

The first issue is called *item exposure*, and refers to the problem that some items might be too often administered with respect to other items. One reason might be that these items are often selected as initial items, or because they are very informative at average ability level (Davey and Parshall 1995; van der Linden 1998). However, allowing such items to be too often administered yields a security problem for the item bank: pre-knowledge about items too often exposed can become available for examinees by gathering related information from previous test takers. Another related problem is an increased cost in developing and calibrating new items to be introduced in the item bank. For those reasons, it is important to ensure that items are not administered too frequently (Chang and Ying 1999; Stocking and Lewis 1998).

To control for item exposure, several methods were suggested. Some rely on the selection of more than one optimal item in the neighbourhood of the current ability estimate, and some random selection is made among these items to select the next administered one. Such methods include the so-called *randomesque* method (Kingsbury and Zara 1989, 1991) and the 5-4-3-2-1 technique (McBride and Martin 1983; Chang and Ansley 2003). More sophisticated methods, based on maximum allowed item exposure rates that are determined by means of prior simulations, include the Sympson and Hetter method (Hetter and Sympson 1997), the Davey and Parshall approach (Davey and Parshall 1995; Parshall, Davey, and Nering 1998) and the Stocking and Lewis conditional and unconditional multinomial methods (Stocking and Lewis 1995, 1998). It is also worth mentioning the recent method of maximum priority index (Cheng and Chang 2009).

The second issue is often referred to as *content balancing* and it consists in selecting items from various subgroups of an existing structure of the item bank. The selection must be such that it is balanced with respect to some predefined percentages of items coming from each subgroup (Kingsbury and Zara 1989). This is particularly usefiul when the item bank has a natural structure in subgroups of targeted goals or competencies, for instance an item bank of mathematics items with a natiral classification into addition, substraction, multiplication and division problems. Forcing the CAT content to be balanced ensures that at least some percentage of the test items come from each subgroup of items.

In sum, controlling for content balancing requires: (a) an intrinsic classification of items into subgroups of targeted areas; (b) a set of relative proportions of items to be administered from each subgroup. With these elements, Kingsbury and Zara (1989) proposed a simple method, sometimes referred to as the *constrained content balancing* method (Leung, Chang, and Hau 2003):

1. At each step of the CAT process, compute the current empirical relative proportions for each subgroup.

2. Determine the subgroup with largest difference between the theoretical relative proportion and its empirical value.

3. Select the next item from this subgroup and return to step 1.

More sophisticated content balancing methods are available; see for instance Leung *et al.* (2003) and Riley, Dennis, and Conrad (2010). Interestingly, the method of maximum priority index (Cheng and Chang 2009) can also be considered for content balancing.

4. The catR package

The main steps of a CAT process have been sketched in the previous section. In this section we provide details about the functionalities of the **catR** package, in relationship with the different CAT steps.

4.1. Structure

The **catR** package is all organized around a single command called `randomCAT`. It generates a response pattern given the true ability level of the examinee, an item bank, the maximal

number of items to be administered, and four lists of input parameters. These lists correspond to the different steps described in Section 2 and are detailed below. The basic R code for `randomCAT` is as follows:

```
randomCAT(theta, bank, maxItems, cbControl, start, test, stop, final)
```

In this code, `theta` is the true ability level; `bank` is an item bank with the appropriate format (see later); `maxItems` is the maximal test length (set to 50 items by default); `cbControl` is an inpout parameter that controls for content balancing (whenever required); and `start`, `test`, `stop` and `final` are four lists of input parameters. These lists are fully described in the forthcoming sections.

Note that to avoid misspecifications of the input lists, **catR** has an integrated function called `testList`, which determines whether the lists have the appropriate format. If at least one of the lists is not correctly specified, `testList` returns an error message with indications on the detected misspecifications.

4.2. Item bank generation

To create an item bank in an appropriate format for the `randomCAT` function, **catR** makes use of the `createItemBank` function. This command has twelve input parameters, some are mandatory, others are optional with default values. These parameters are listed in Table 2, together with their role, possible values, default values, and the cases where they are ignored by the function.

The starting point to generate an item bank is the matrix of item parameters. This matrix must have one row per item, and at least four columns. The columns hold respectively the values of the item discriminations, difficulties, pseudo-guessing and inattention parameters, in this order. This matrix can be passed through the `items` argument. A fifth column can be added to the matrix, holding the names of the subgroups of items for content balancing. In this case, the full matrix can be passed as a data frame. If this fifth column is absent then content balancing cannot be controlled. Note that to allow for content balancing control, the argument `cb` must be additionally set to `TRUE` (default value is `FALSE`), otherwise the fifth column is discarded for creating the item bank.

Alternatively, the user can let the software generate the item parameters. In this case, the `items` argument contains the number of items to be included in the bank. Furthermore, the argument `model` must be set to determine which IRT model is used to generate item parameters. Possible values of model are `"1PL"`, `"2PL"`, `"3PL"` and `"4PL"`, with reference to each of the four logistic models. By default, the `"4PL"` model is considered, and the following distributions are used: $N(0, 1)$ for item difficulties, $N(1, 0.04)$ for item discriminations, $U(0, 0.25)$ for item pseudo-guessing parameters, and $U(0.75, 1)$ for item inattention parameters.

These prior distributions can be modified by setting optional arguments `aPrior`, `bPrior`, `cPrior` and `dPrior` accurately. These arguments take as values, a vector of three components, the first one coding for the distribution name, and the last two holding the distribution parameters. For item discriminations, available distributions are the normal, the log-normal, and the uniform densities. For item difficulties, both normal and uniform distributions are currently available. Finally, for both pseudo-guessing and inattention parameters, the Beta and the uniform distributions are possible values (see the help file of `createItemBank` for further details).

Argument	Role	Value	Default	Ignored if
`items`	fixes the number of items to create, or provides the item parameter values	an integer value or a matrix (or data frame)	`NA`	///
`cb`	should content balancing be controlled?	`TRUE` or `FALSE`	`FALSE`	`items` is an integer
`model`	specifies the IRT model for item parameter generation	`"1PL"`, `"2PL"`, `"3PL"` or `"4PL"`	`"4PL"`	`items` is a matrix
`aPrior`	specifies the prior distribution of item discriminations	a vector with distribution components	`c("norm", 1, 0.2)`	`items` is a matrix
`bPrior`	specifies the prior distribution of item difficulties	a vector with distribution components	`c("norm", 0, 1)`	`items` is a matrix
`cPrior`	specifies the prior distribution of item pseudo-guessing	a vector with distribution components	`c("unif", 0, 0.25)`	`items` is a matrix
`dPrior`	specifies the prior distribution of item inattention	a vector with distribution components	`c("unif", 0.75, 1)`	`items` is a matrix
`seed`	fixes the seed for random generations	a real value	`1`	`items` is a matrix
`thMin`	fixes the minimum ability value for information grid	a real value	`-4`	///
`thMax`	fixes the maximum ability value for information grid	a real value	`4`	///
`step`	fixes the step between ability values for information grid	a positive real value	`0.01`	///
`D`	fixes the constant metric	a positive real value	`1`	`items` is a matrix

Table 2: Arguments of the `createItemBank` function.

For simpler models, the corresponding parameters are constrained accordingly; for instance, under the 2PL model, the third and fourth columns will be all zeros and ones, respectively. The `D` sets the constant metric, by default 1 for the logistic metric. The other common choice is to set `D` to 1.7 for the normal metric (Haley 1952). Arguments `model`, `aPrior`, `bPrior`, `cPrior`, `dPrior`, `seed` and `D` are ignored if the user provides a matrix (or data frame) through the `items` argument.

Once the matrix of item parameters is available, an item information matrix is created. This matrix contains the values of the item information function for each item in the bank, and for a set of predefined ability levels. These abilities are chosen from a sequence of values entirely defined by the argument `thMin`, `thMax` and `step`. The arguments display respectively: the minimum ability level, the maximum ability level, and the step value between two abilities of the sequence. The default values are `-4`, `4` and `0.01`, which means that the default information matrix refers to ability levels from -4 to 4 by steps of 0.01. Each row of the information

Argument	Role	Value	Default	Ignored if
`fixItems`	specifies the items to be administered	`NULL` or a vector of items	`NULL`	///
`seed`	fixes the seed for the random selection of items	`NULL` or a real value	`NULL`	`fixItems` is not `NULL`
`nrItems`	fixes the number of items to be administered	an integer value value	1	`fixItems` is not `NULL` not `NULL`
`theta`	fixes the centre of the range of abilities	a real value	0	`fixItems` or `seed` is not `NULL`
`halfRange`	fixes the bandwidth of the range of abilities	a positive real value	4	`fixItems` or `seed` is not `NULL`
`startSelect`	specifies the method for item selection	`"bOpt"` or `"MFI"`	`"bOpt"`	`fixItems` or `seed` is not `NULL`

Table 3: Arguments of the `start` list for the initial step.

matrix refers to one ability level, and each column to one of the bank items.

Note that for a correct application of `randomCAT`, the item bank must be generated prior to starting the CAT process. It is therefore recommended to generate the item bank with the `createItemBank` function and to save it into an R object. This object can then be used in any further application of `randomCAT`.

4.3. Initial step

The initial step, that is, the selection of the first item(s) to be administered, is coded by the `start` list in the `randomCAT` function. Table 3 summarizes the six possible arguments of this list, together with additional information such as their default values and the situations where they are ignored.

The three methods for selecting the first item(s) are: by user specification, by random selection, or by optimal selection according to some fixed constraints. The three methods are set up as follows.

First, the user specification of the first items is done through the `fixItems` argument. It is a vector of integer values coding for the item numbers, in the order they are listed in the item bank. By default, `fixItems` takes the `NULL` value, which means that the user does not select the first items. Then, one can choose these first items randomly. Two arguments must be specified: `seed` takes a numeric value that fixes the random seed for item selection, and `nrItems` that indicates how many items must be randomly chosen. If `seed` is `NULL` (which is the default value), random selection is ignored, and the third method applies. This third method, the optimal selection of the starting items, relies on two aspects. The first one is the subset of ability levels that will be used for optimal item selection. Each value in this subset corresponds to one selected item. The subset is defined by fixing an interval of ability levels and by splitting this interval into a subset of equidistant values. The interval is set up by its centre, through the argument `theta`, and by its half-range (i.e., half the range of the interval), through the `halfRange` argument. The number of values is still given by the `nrItems`

argument. For instance, the subset $-1, 0, 1$ of ability values is set up by the triplet of values $(3, 0, 1)$ for the triplet of arguments (`nrItems`, `theta`, `halfRange`). As another example, the subset $(-1.5, 0, 1.5, 3)$ is given by $(4, 0.75, 2.25)$. A single item can also be selected: the triplet of values (1, `th`, `hr`) reduces the subset to the value `th`, and the value `hr` of the half-range is irrelevant.

When this subset of ability values is fixed, the second aspect of the optimal item selection method is the criterion used for selecting these items. Two possible methods are implemented, through the `startSelect` argument: Urry's rule and the information function rule. Urry's rule, coded by `startSelect = "bOpt"`, selects the items whose difficulty levels are as close as possible to the ability values in the subset. The information function rule, coded by `startSelect = "MFI"`, selects the items that maximize the item information function at the given ability levels. By default, items are selected according to their difficulty level.

It is important to notice that the three methods are hierarchically ordered. If `fixItems` is not `NULL`, items will be chosen according to the user's pre-specifications, and all other arguments of the start list will be ignored. If `fixItems` is `NULL` and `seed` is not `NULL`, then `nrItems` will be randomly selected according to the `seed` value, and all other arguments are again ignored. Finally, if both `fixItems` and `seed` are `NULL`, then the optimal selection of the first items is achieved, according to the values of the `nrItems`, `theta`, `halfRange` and `startSelect` arguments. Note also that not all the arguments need be specified. For instance, if only `seed` is specified, then a single item will be randomly chosen in the item bank, as the default value of `nrItems` is one.

4.4. Test step

The test step is specified in `randomCAT` by means of the `test` list, which contains at most nine arguments as listed in Table 4. These arguments refer to the selected method of provisional ability estimation and the rule for next item selection, as well as for item exposure control.

First, the `method` argument fixes the estimation method. The possible values are `"ML"`, `"BM"`, `"EAP"` and `"WL"` for the respective methods, and the default choice is `"BM"`. If the method is either Bayesian modal (`"BM"`) or expected a posteriori (`"EAP"`), the `priorDist` and `priorPar` arguments determine the prior distribution. The argument `priorDist` fixes the distribution itself, while `priorPar` specifies the related prior parameters.

Three cases can arise. First, `priorDist` takes the (default) value `"norm"`, corresponding to the normal distribution, and `priorPar` is a vector of two numeric components, the prior mean and the prior standard deviation. Second, `priorPar` takes the value `"unif"` for the uniform distribution, and `priorPar` is a vector of two numeric components for the range of the uniform distribution. Third, `priorPar` takes the value `"Jeffreys"` for Jeffreys' prior distribution. In this case, the `priorPar` argument is ignored.

Moreover, for EAP estimation, it is possible to set some parameters for the numerical approximation of the integrals. These parameters are the limits for adaptive quadrature and the number of quadrature points, and are provided altogether through the `parInt` argument. The default value of `parInt` is the vector `c(-4, 4, 33)`, specifying 33 quadrature points on the range $[-4, 4]$, that it, the sequence from -4 to 4 by steps of 0.25.

The next two arguments are `D` and `range`. The argument `D` fixes the metric value (see Table 2), and the argument `range` sets the range of allowable ability estimates. Its primary use is to avoid infinite estimates. The default range is $[-4, 4]$ and can be changed by providing the

Argument	Role	Value	Default	Ignored if
`method`	specifies the method for ability estimation	`"BM"`, `"ML"` `"EAP"` or `"WL"`	`"BM"`	///
`priorDist`	specifies the prior distribution	`"norm"`, `"unif"` or `"Jeffreys"`	`"norm"`	`method` is neither `"BM"` nor `"EAP"`
`priorPar`	specifies the parameters of the prior distribution	a vector of two real values	`c(0, 1)`	`method` is neither `"BM"` nor `"EAP"`, or `priorDist` is `"Jeffreys"`
`range`	fixes the maximal range of ability values	a vector of two real values	`c(-4, 4)`	`method` is `"EAP"`
`D`	fixes the value of the metric constant	a positive real value	1	///
`parInt`	fixes the parameters for numerical integration	a vector of three numeric values	`c(-4, 4, 33)`	`method` is not `"EAP"`
`itemSelect`	specifies the method for next item selection	`"MFI"`, `"MEPV"`, `"MEI"`, `"MLWI"`, `"MPWI"`, `"Urry"` or `"random"`	`"MFI"`	///
`infoType`	specifies the type of information function	`"observed"` or `"Fisher"`	`"observed"`	`itemSelect` is not `"MEI"`
`randomesque`	specifies the number of optimal items for 'randomesque' item exposure	a positive integer	1	///

Table 4: Arguments of the `test` list for the test step.

bounds of the interval through the `range` argument.

The `itemSelect` argument specifies the method for selecting the next item. Currently seven methods are available: MFI, MEI, MEPV, MLWI, MPWI, Urry's rule and random selection. The first five methods are set up by their acronym (i.e., `"MFI"` for MFI method etc.), Urry's rule by the value `"Urry"` and random selection by `"random"` value. The default method is the MFI criterion. In addition, the type of information function, either observed or Fisher, can be defined by the `infoType` argument. The two possible values are `"observed"` and `"Fisher"` and the default value is `"observed"`. Note that it is only useful for MEI criterion, and is ignored with all other methods.

Finally, item exposure can be controlled with the *randomesque* approach (Kingsbury and Zara 1989). It consists in selecting not only one but several optimal items, according to the specified criterion, and to randomly pick-up one of these optimal items for the next step of the CAT process. This is controlled by the argument `randomesque`, taking a positive integer value, with default value 1 (that is, the usual selection of the optimal item). This is currently the only available method for item exposure control within **catR**.

Argument	Role	Value	Default	Ignored if
`rule`	specifies the stopping rule	`"length"`, `"precision"` or `"classification"`	`"length"`	///
`thr`	specifies the threshold related to the stopping rule	a real value	20	///
`alpha`	specifies the alpha level for the provisory confidence intervals	a real value	0.05	`rule` is not `"classification"`

Table 5: Arguments of the `stop` list for the stopping step.

4.5. Stopping step

The stopping step is defined by the `stop` list. This list has at most three arguments, which are listed in Table 5.

The first argument, `rule`, specifies the type of stopping rule. Three values are possible: `"length"`, `"precision"` and `"classification"`, each referring to a particular rule as described in Section 3.3. The default rule is the `"length"` criterion. The second argument is called `thr` and holds the specific threshold of the stopping rule. The values of `thr` are either: an integer fixing the test length (for the length criterion), a real positive value giving the maximum allowable standard error of the ability estimate (for the precision criterion), or the value of the ability level to be considered for subject classification (for the classification criterion). Because the default stopping rule is the length criterion, the default thr value is 20, that is, the CAT process will stop by default after 20 items are administered.

The last argument is called `alpha` and it permits to set the confidence level of the intervals for the classification criterion. More precisely, the confidence level is one minus `alpha`, and the default `alpha` level is 0.05. This argument is obviously ignored if the rule is either `"length"` or `"precision"`.

It is important to notice that the stopping rule might not be satisfied if the `stop` list is badly specified. For instance, setting a very small standard error for the precision criterion would maybe require too many items, so that the CAT process becomes useless. Another problem would arise with the classification criterion, if either the threshold or the significance level is misspecified, so that too many items are also required to classify the subject. In order to avoid such an issue, `randomCAT` has a "meta-argument", called `maxItems`, which imposes a maximal test length and forces the test to stop if this length is reached, even though the stopping rule is not yet satisfied. In this case, the output will display a warning message reminding that "the stopping rule was not satisfied". Note that if the length criterion is chosen, then the test length is taken as the minimum of the `maxItems` meta-argument and the `thr` argument of the `stop` list. The default value of `maxItems` is 50.

4.6. Final step

The `final` list sets the ability estimation method for the final step of the CAT process, with two exceptions. First, the `itemSelect` and `infoType` arguments are useless for the final step,

and are therefore not allowed. Second, the `final` list has an additional argument, `alpha`, which fixes the significance level for the final confidence interval of the ability level. This `alpha` argument performs similarly to that of the `stop` list (see Table 5), and takes the default value 0.05. Note also that the final confidence interval is always computed, even if the stopping rule is not the classification criterion. All other arguments of the `final` list are completely identical to those of the `test` list. However, they may take different values, so that the provisional and final ability estimators are possibly different.

4.7. Content balancing

If the item bank was specifically created for content balancing, the latter can be controlled through the argument `cbControl` of the `randomCAT` function. The only content balancing method currently available in **catR** is the constrained content balancing method (Kingsbury and Zara 1989, 1991). By default, `cbControl` takes the `NULL` value, so nothing is performed. Otherwise, `cbControl` must be set as a list of two elements called `names` and `props`. The element `cbControl$names` contains the names of the subgroups of items, while `cbControl$props` holds the expected relative proportions of items per subgroup. For compatibility issues, the names of the subgroups must exactly match the subgroups' names provided in the item bank, and the relative proportions must be non-negative (they may possibly not sum to one; in this case they are internally normalized to sum to one). Note that **catR** has an internal function called `test.cbList` that tests the format of the `cbControl` list, and returns an error message if the list is not accurate for content balancing.

4.8. Output

The structure of an output from `randomCAT` is a list of class `"cat"` with many arguments. Basically, the function returns the final results: final ability estimation, standard error and confidence interval. Moreover, the complete response pattern and the matrix of item parameters of selected test items are also returned. For completeness, the vectors of provisional ability estimates and standard errors are displayed, so that one can plot the curve of ability estimation through the CAT process. Finally, all input arguments of the `start`, `test`, `stop` and `final` lists are returned within the `randomCAT` output, as well as the value of the `cbControl` argument. The complete list of argument names is described in the **catR** help manual.

Two additional functions are useful for a simpler and more suitable display of the CAT results. First, the `print` function for `"cat"` objects organizes the output and displays it in a visually attractive way (see next section for an example). The printing of the results include: a summary of the basic options for the selection of early test items, the provisional ability estimator, the stopping rule, the control for item exposure etc. Then, the different steps of the CAT process are displayed in a table format, with the administered item, the response to this item, and the updated provisional estimate of ability and standard error. Next, the final results are displayed: final ability estimator, final estimate, standard error and confidence interval. If content balancing was controlled, additional output is also displayed: a summary table of both expected and observed relative proportions of items administered per subgroup, and the list of items administered per subgroups.

The second utility function is the `plot` command for `"cat"` objects. It permits a visual representation of the CAT process by plotting the provisional ability levels against the test

length. The function can be used as follows:

```
plot(x, ci = FALSE, alpha = 0.05, trueTh = TRUE, classThr = NULL)
```

In this code, `x` is the output of the `randomCAT` function; `ci` is a logical argument that specifies whether or not confidence intervals must be drawn around each provisional ability estimate; and `alpha` is the significance level for building the confidence intervals (0.05 by default). In addition, `trueTh` is a logical value specifying whether the true ability level must be drawn additionally to the plot (as a horizontal line), and `classThr` is the value of the classification threshold to be drawn in addition (the value `NULL` disables this argument). Only the `x` argument is mandatory; by default, the set of provisional and final ability estimates are drawn, without each confidence interval, and the true ability level is superposed to the plot.

5. Examples

We illustrate the package by creating two item banks and by generating response patterns under pre-specified CAT processes. In the first example, a large item bank is considered and the main goal of the test is to discriminate between very good examinees and the others. In the second example, a real item bank is taken into account and both item exposure and content balancing are under control. Because the response patterns are drawn by a random process, the following output may differ from two successive runs of the same R code.

5.1. Example 1

The item bank is made of 500 items whose parameters are randomly generated under a 3PL model. The information matrix uses the sequence of ability levels from -4 to 4 by steps of 0.05. The R code for creating this item bank is given below (both the random seed and the constant metric `D` are kept to their default values of 1 and the resulting item bank is stored into the so-called `Bank` object):

```
R> Bank <- createItemBank(items = 500, model = "3PL", thMin = -4, thMax = 4,
+    step = 0.05)
```

Now, the four lists of tuning parameters are defined as follows. To start the test, a single item will be chosen such that it maximizes Fisher information function at the ability level 0. This is to start the test by administrating one average difficulty item to the examinee, which is a common starting situation in CAT. The corresponding list is stored into the `Start` object:

```
R> Start <- list(nrItems = 1, theta = 0, startSelect = "MFI")
```

Note that neither `fixItems` nor `seed` are specified as they take the value `NULL` by default.

For the test step, the WL estimator is chosen, and the usual MFI criterion is considered for next item selection. This is summarized by the following R code, and the corresponding list is stored into the `Test` object:

```
R> Test <- list(method = "WL", itemSelect = "MFI")
```

The selected stopping rule is the classification rule, and it is decided that the CAT process should stop when the ability level is significantly different from the value 2 with a confidence level of 95%. This situation depicts a potential CAT process wherein one wishes to discriminate between very good examinees with ability levels larger than 2, and the other examinees. The following R code is suitable for this rule, by storing the corresponding list to the `Stop` object:

```
R> Stop <- list(rule = "classification", thr = 2, alpha = 0.05)
```

When the adaptive test ends, the final ability estimate is obtained by weighted likelihood estimation, as during the adaptive test. Though it is not mandatory to keep the same ability estimator for both test and final steps, this is often the case in practice. Moreover, the 95% confidence interval for ability is built up using the final standard error of the WL estimator. This is summarized by the following code:

```
R> Final <- list(method = "WL", alpha = 0.05)
```

The four steps of the CAT process are now plugged in the `randomCAT` function as given below. For the particular simulation to discuss later on, one examinee with true ability level equal to 1 is under examination. It is expected that the examinee will be classified as having an ability level lower than 2 with a rather short adaptive test. The maximal number of items to be administered is kept to 50, the default value. The result of this random CAT generation is stored into the R object `res` and displayed below:

```
R> res <- randomCAT(trueTheta = 1, itemBank = Bank, start = Start,
+    test = Test, stop = Stop, final = Final)
R> res

Random generation of a CAT response pattern

 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: maximum Fisher information
   Provisional ability estimator: Weighted likelihood estimator

 Stopping rule:
   Stopping criterion: classification based on 95% confidence interval
   Classification threshold: 2
   Maximum test length: 50 items
```

```
 Item exposure control:
   Method: 'randomesque'
   Number of 'randomesque' items: 1

 Content balancing control:
   No control for content balancing

 Adaptive test details:

Nr        1     2     3     4     5     6     7     8     9    10    11
Item    343   337   170   479   316   239   420    70   362   385   421
Resp.     1     0     1     1     1     1     1     0     0     1     0
Est.  0.884 0.553 0.802 1.261 1.508  1.72 1.972  1.79 1.544 1.673 1.504
SE    1.023 0.827 0.745 0.722 0.693 0.683 0.683  0.61 0.549 0.535 0.499

Nr       12    13    14    15    16
Item    341   464   210    83   393
Resp.     0     1     1     0     0
Est.   1.26 1.329 1.391 1.254 1.137
SE    0.467 0.453 0.442 0.422 0.407

 Final results:
   Length of adaptive test: 16 items
   Final ability estimator: Weighted likelihood estimator
   Final ability estimate (SE): 1.137 (0.407)
   95% confidence interval: [0.34,1.933]
   Final subject classification: ability is smaller than 2
```

The first part of the output is a summary of the different settings of the test step. The true ability level is mentioned first, followed by the parameters for the initial step, the test step, and the stopping rule. Item exposure control is then described; currently only the "randomesque" method is available, and in this example only one optimal item is selected. That is, there is no control of item exposure. Also, no control for content balancing was performed, which is also indicated in the output.

The details of the adaptive test are then returned. It consists in a matrix with one column per administered item, and five rows with respectively the item number of administration, the item identification number (as listed in the item bank), the generated response to this item, the provisional ability estimate and its standard error. This output ends with the results of the final step: final ability estimator, final estimate and standard error, final confidence interval and the final conclusion with respect to the classification rule.

In the simulated example, 16 items were administered before the conditions of the stopping rule were met. As the number of items increases, the standard error of ability estimate decreases, as the amount of available information becomes larger as the test goes on. After 16 items, the ability estimate is close to 1, the true ability level, and the standard error is sufficiently small to classify the examinee as having an ability smaller than 2, with 95% confidence. Indeed, the final 95% confidence interval is [0.340; 1.933] and does not contain the

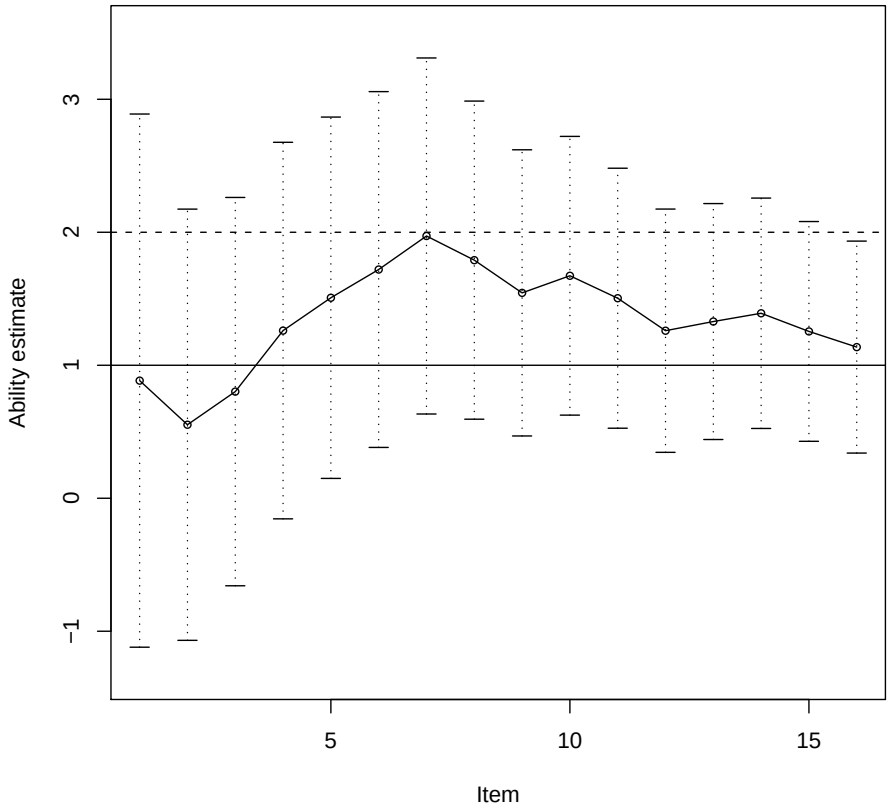

Figure 2: Plot of the results of the CAT simulation. The classification threshold is displayed by the horizontal dashed line and the true ability level by a solid horizontal line.

classification threshold 2. Note that, on the other hand, this interval covers the true ability level.

The provisional ability estimates can be graphically displayed by using the `plot.cat` function. In addition to each point estimate, provisional 95% confidence intervals are also drawn, together with the classification threshold of 2 (dashed line) and the true ability level of 1 (solid line). This is summarized with the code:

```
R> plot(res, ci = TRUE, trueTh = TRUE, classThr = 2)
```

and the output plot is given in Figure 2. As the test goes on, the available information increases and the provisional standard errors of the ability estimates decrease accordingly. Consequently, narrower confidence intervals are obtained. After the 16th item administration, the confidence interval does not contain the threshold value 2 anymore, which forces the CAT process to stop. This is obviously in agreement with the R output above, but is visually displayed in this plot. Note also that the ability estimates are getting closer to the true level 1 as the items are being administered.

5.2. Example 2

In the second example, the TCALS item bank is considered. The TCALS is an English skill assessment test assigned to students entering into college studies in the Canadian province of Quebec (Laurier, Froio, Pearo, and Fournier 1998). The test is made of 85 items, calibrated under the 3PL model (Raiche 2002), and can be merged into five categories: two categories of oral items (`Audio1` and `Audio2`) and three categories of written items (`Written1`, `Written2` and `Written3`). This item bank is available from **catR** directly; see the corresponding help file for further details.

The generated CAT process has the following characteristics:

1. Five initial items are selected as the most infomrative at target ability levels from -2 to 2 by steps of 1.
2. Ability level is estimated with the EAP estimator and prior standard normal distribution, both during the test and at the final step.
3. The criterion for next item selection is Urry's rule.
4. The CAT process stops when 12 items are administered.
5. Item exposure is controlled by selecting, at each step of the CAT process, the four most optimal items (i.e., the "randomesque" number of items is four).
6. Control for content balancing is imposed to the five subgroups (`Audio1`, `Audio2`, `Written1`, `Written2`, `Written3`) by the following relative proportions: (0.1, 0.2, 0.2, 0.2, 0.3).

First, the item bank must be created appropriately:

```
R> data("tcals")
R> Bank2 <- createItemBank(tcals, cb = TRUE)
```

Then, the four steps of the CAT are further defined:

```
R> Start2 <- list(nrItems = 5, theta = 0, halfRange = 2)
R> Test2 <- list(method = "EAP", itemSelect = "Urry", randomesque = 4)
R> Stop2 <- list(rule = "length", thr = 12)
R> Final2 <- list(method = "EAP")
```

The control for content balancing is performed by creating the following list:

```
R> cbList <- list(names = c("Audio1", "Audio2", "Written1", "Written2",
+    "Written3"), props = c(0.1, 0.2, 0.2, 0.2, 0.3))
```

Finally, the CAT process is run for a true ability level of zero, with the previous settings:

```
R> res2 <- randomCAT(trueTheta = 0, itemBank = Bank2, start = Start2,
+    test = Test2, stop = Stop2, final = Final2, cbControl = cbList)
R> res2
```

and the output is:

```
Random generation of a CAT response pattern

 True ability level: 0

 Starting parameters:
   Number of early items: 5
   Early items selection: matching item difficulties to starting abilities
   Starting abilities: -2, -1, 0, 1 and 2
   Order of starting abilities administration: 2, 1, -1, 0 and -2

 Adaptive test parameters:
   Next item selection method: Urry's procedure
   Provisional ability estimator: Expected a posteriori (EAP) estimator
   Provisional prior ability distribution: N(0,1) prior

 Stopping rule:
   Stopping criterion: length of test
   Maximum test length: 12 items

 Item exposure control:
   Method: 'randomesque'
   Number of 'randomesque' items: 4

 Content balancing control:
   Expected proportions of items per subgroup:

 Audio1 Audio2 Written1 Written2 Written3
 0.1    0.2    0.2      0.2      0.3


 Adaptive test details:

Nr        1     2     3     4     5     6     7     8     9    10    11
Item     77    25    67    24    38    10    61    63    45    80    44
Resp.     1     1     1     0     1     1     0     1     1     0     1
Est.     NA    NA    NA    NA 0.343 0.571 0.032  0.31  0.34  0.23 0.248
SE       NA    NA    NA    NA 0.718 0.593 0.495 0.416 0.398 0.355 0.342

Nr       12
Item     76
Resp.     0
Est.   0.16
SE    0.336

 Final results:
   Length of adaptive test: 12 items
   Final ability estimator: Expected a posteriori (EAP) estimator
```

```
   Final prior distribution: N(0,1) prior
   Final ability estimate (SE): 0.16 (0.336)
   95% confidence interval: [-0.498,0.818]
   Proportions of items per subgroup (expected and observed):

     Audio1 Audio2 Written1 Written2 Written3
Exp. 0.100  0.200  0.200    0.200    0.300
Obs. 0.083  0.167  0.250    0.167    0.333

   Items administered per subgroup:

   Audio1: 10
   Audio2: 24, 25
   Written1: 38, 44, 45
   Written2: 61, 63
   Written3: 67, 76, 77, 80
```

The output is similar to the previous example, and up to the modifications of the CAT design (listed in the top of the output) and the output of this particular simulated run, the main two differences concern the item exposure and content balancing output. First, item exposure is controlled by selecting the four optimal items at each step, which is returned under the `Item exposure control` section. Second, the expected relative proportions of items per subgroup are printed, as input information for the CAT process and because content balancing control was required.

Finally, at the end of the output, two pieces of additional information are printed. First, a two-row table is printed; it contains both the expected and the observed (empirical) relative proportions of items per subgroup. This is an easy way to check the discrepancy between the expected content balancing and the one that was actually obtained after the test. The second information is a merging of the test items into the five subgroups of items, with their names. This permits to extract straightforward information about which items of which type were administered.

6. Discussion

This paper describes an R package as a technical tool for generating CAT response patterns. Dichotomous logistic item response models are used to generate such response patterns. Several criteria for selecting the first items of the test, for selecting the next item, for stopping the adaptive test, and for final ability estimation, are available. Input options reflect a wide variety of real situations, and the output information is complete and easy to extract. For those reasons, **ctaR** can be used as is for simulation studies with the R environment, or as part of more sophisticated software with user interface (similarly as **Firestar** software).

The main assets of **catR** are the flexibility with respect to the logistic IRT model (to our knowledge, it is the first CAT software incorporating the 4PL model), the selection of the first items of the test and the efficiency to generate a large number of CAT response patterns. It is comparable to **Firestar** with respect to the methods for next item selection and item exposure, to **CATSim** with respect to the stopping rules, and to both software with respect

to the selection of ability estimators. In addition, control for content balancing seems to be an asset of **catR** with respect to other software. Its main drawback, however, is that the current version of **catR** is limited to dichotomous items, whereas both **Firestar** and **CATSim** can handle polytomously scored items. The polytomous version of **catR** with Bock's nominal response model (Bock 1972) as underlying IRT model, is currently under development. In addition, other methods for item exposure (Chang and Ansley 2003) and content balancing (Leung *et al.* 2003; Riley *et al.* 2010) could be implemented.

In addition to the presentation of the package itself, the main objective of this paper is to provide a clear and simple overview of the CAT framework. The four steps of this method (initial step, test step, stopping rule, final step) are discussed according to the current standards and technical accuracy. The primary intention is to provide the reader a short but straightforward introduction to the CAT environment, with suggested references for further reading to the interested reader. Wainer (2010) is claiming that one should focus further on the possibilities that CAT can offer with respect to standard, non adaptive, assessment tests. We believe this paper is one small step towards this direction, both on the methodological and the practical aspects.

Interestingly, **catR** was already combined with the web-based platform **Concerto** (Kosinski and Rust 2011a,b) to create web-based adaptive tests. Although still in development, **Concerto** would eventually permit any interested researcher or CAT user to build specific CAT processes, using the web platform for item bank creation and **catR** as underlying package for routine calculations. See `http://code.google.com/p/concerto-platform/` for further information.

Computational details

The currently available version of **catR** is 2.0. Version 2.12.0 (or later) of the R software should be installed for optimal working of **catR**.

Acknowledgments

The authors wish to thank Florian Wickelmaier, Carolin Strobl, Achim Zeileis, and two anonymous referees for their helpful suggestions. This research was financially supported by a postdoctoral grant "Chargé de recherches" of the National Funds for Scientific Research (FNRS), Belgium, the Research Funds of the KU Leuven, Belgium, and the Social Science and Humanities Research Council of Canada (SSHRC).

References

Andersen EB (1970). "Asymptotic Properties of Conditional Maximum Likelihood Estimators." *Journal of the Royal Statistical Society B*, **32**, 283–301.

Andersen EB (1972). "The Numerical Solution of a Set of Conditional Estimation Equations." *Journal of the Royal Statistical Society B*, **34**, 42–54.

Assessment Systems Corporation (2012). ***CATSim**: Comprehensive Simulation of Computerized Adaptive Testing*. St. Paul, MN. URL `http://www.assess.com/`.

Barton MA, Lord FM (1981). "An Upper Asymptote for the Three-Parameter Logistic Model." *Technical Report RR-81-20*, Educational Testing Service, Princeton, NJ.

Birnbaum A (1968). "Some Latent Trait Models and Their Use in Inferring an Examinee's Ability." In FM Lord, MR Novick (eds.), *Statistical Theories of Mental Test Scores*, pp. 395–479. Addison-Wesley, Reading.

Birnbaum A (1969). "Statistical Theory for Logistic Mental Test Models with a Prior Distribution of Ability." *Journal of Mathematical Psychology*, **6**, 258–276.

Bock RD (1972). "Estimating Item Parameters and Latent Ability when Responses Are Scored in Two or More Nominal Categories." *Psychometrika*, **37**, 29–51.

Bock RD, Aitkin M (1981). "Marginal Maximum Likelihood Estimation of Item Parameters. An Application of the EM Algorithm." *Psychometrika*, **37**, 29–51.

Bock RD, Mislevy RJ (1982). "Adaptive EAP Estimation of Ability in a Microcomputer Environment." *Applied Psychological Measurement*, **6**, 431–444.

Chang H, Ying Z (1999). "*a*-Stratified Multistage Computerized Adaptive Testing." *Applied Psychological Measurement*, **23**, 211–222.

Chang SW, Ansley TN (2003). "A Comparative Study of Item Exposure Control Methods in Computerized Adaptive Testing." *Journal of Educational Measurement*, **40**, 71–103.

Cheng Y, Chang HH (2009). "The Maximum Priority Index Method for Severely Constrained Item Selection in Computerized Adaptive Testing." *British Journal of Mathematical and Statistical Psychology*, **62**, 369–383.

Choi SW (2009). "**Firestar**: Computerized Adaptive Testing Simulation Program for Polytomous Item Response Theory Models." *Applied Psychological Measurement*, **33**, 644–645.

Choi SW, Swartz RJ (2009). "Comparison of CAT Item Selection Criteria for Polytomous Items." *Applied Pscyhological Measurement*, **32**, 419–440.

Davey T, Parshall CG (1995). "New Algorithms for Item Selection and Exposure Control with Computerized Adaptive Testing." Paper presented at the Annual Meeting of the American Educational Research Association, San Francisco, CA.

Drasgow F, Olson-Buchanan JB (1999). *Innovations in Computerized Assessment*. Lawrence Erlbaum Associates, Hillsdale.

Georgiadou EG, Triantafillou E, Economides AA (2007). "A Review of Item Exposure Control Strategies for Computerized Adaptive Testing Developed from 1983 to 2005." *Journal of Technology, Learning, and Assessment*, **5**, 1–38.

Green BF (2011). "A Comment on Early Student Blunders on Computer-Based Adaptive Tests." *Applied Psychological Measurement*, **35**, 165–174.

Haley DC (1952). "Estimation of the Dosage Mortality Relationship when the Dose Is Subject to Error." *Technical Report no 15*, Applied Mathematics and Statistics Laboratory, Stanford University, Palo Alto, CA.

Hetter RD, Sympson JB (1997). "Item Exposure Control in CAT-ASVAB." In WA Sands, BK Waters, JR McBride (eds.), *Computerized Adaptive Testing: From Inquiry to Operation*, pp. 141–144. American Psychological Association, Washington, DC.

Hoijtink H, Boomsma A (1995). "On Person Parameter Estimation in the Dichotomous Rasch Model." In GH Fischer, IW Molenaar (eds.), *Rasch Models. Foundations, Recent Developments, and Applications*, pp. 53–68. Springer-Verlag.

Irvine SH, Kyllonen PC (2002). *Item Generation for Test Development*. Lawrence Erlbaum Associates, Mahwah.

Jeffreys H (1939). *Theory of Probability*. Oxford University Press, Oxford.

Jeffreys H (1946). "An Invariant Form for the Prior Probability in Estimation Problems." *Proceedings of the Royal Society of London A*, **186**, 453–461.

Kingsbury GG, Zara AR (1989). "Procedures for Selecting Items for Computerized Adaptive Testing." *Applied Measurement in Education*, **2**, 359–375.

Kingsbury GG, Zara AR (1991). "A Comparison of Procedures for Content-Sensitive Item Selection in Computerized Adaptive Tests." *Applied Measurement in Education*, **4**, 241–261.

Kosinski M, Rust J (2011a). "The Development of **Concerto** : An Open Source Online Adaptive Testing Platform." Paper presented at the International Association for Computerized and Adaptive Testing, Pacific Grove, CA.

Kosinski M, Rust J (2011b). "The Development of **Concerto** : An Open Source Online Adaptive Testing Platform." Paper presented at the International Meeting of the Psychometric Society, Hong Kong.

Laurier M, Froio L, Pearo C, Fournier M (1998). "Test de classement d'anglais langue seconde au collégial." *Technical report*, College de Maisonneuve, Montreal, QC.

Leung CK, Chang HH, Hau KT (2003). "Computerized Adaptive Testing: A Comparison of Three Content Balancing Methods." *Journal of Technology, Learning, and Assessment*, **2**, 1–13.

Loken E, Rulison KL (2010). "Estimation of a Four-Parameter Item Response Theory Model." *British Journal of Mathematical and Statistical Psychology*, **63**, 509–525.

Lord FM (1980). *Applications of Item Response Theory to Practical Testing Problems*. Lawrence Erlbaum Associates, Hillsdale.

Lord FM (1983). "Unbiased Estimators of Ability Parameters, of their Variance, and of their Parallel-Forms Reliability." *Psychometrika*, **48**, 233–245.

Lord FM (1984). "Maximum Likelihood and Bayesian Parameter Estimation in Item Response Theory." *Technical Report No RR-84-30-ONR*, Educational Testing Service, Princeton, NJ.

Magis D, Raiche G (2011). "**catR**: An R Package for Computerized Adaptive Testing." *Applied Psychological Measurement*, **35**, 576–577.

McBride JR, Martin JT (1983). "Reliability and Validity of Adaptive Ability Tests in a Military Setting." In DJ Weiss (ed.), *New Horizons in Testing: Latent Trait Test Theory and Computerized Adaptive Testing*, pp. 223–236. Academic Press, New York.

Meijer RR, Nering ML (1999). "Computerized Adaptive Testing: Overview and Introduction." *Applied Psychological Measurement*, **23**, 187–194.

Parshall CG, Davey T, Nering ML (1998). "Test Development Exposure Control for Adaptive Testing." Paper presented at the annual meeting of the National Council on Measurement in Education, San Diego, CA.

Parshall CG, Spray JA, Kalohn JC, Davey T (2002). *Practical Considerations in Computer-Based Testing*. Springer-Verlag.

Raiche G (2002). "Le dépistage du sous-classement aux tests de classement en anglais, langue seconde, au collégial [The Detection of Under-Classification to the Collegial English as a Second Language Placement Tests]." *Technical report*, College de l'Outaouais, Gatineau, QC.

Raiche G, Blais JG (2006). "SIMCAT 1.0 – A SAS Computer Program for Simulating Computer Adaptive Testing." *Applied Psychological Measurement*, **30**, 60–61.

R Development Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Riley BB, Dennis ML, Conrad KJ (2010). "A Comparison of Content-Balancing Procedures for Estimating Multiple Clinical Domains in Computerized Adaptive Testing: Relative Precision, Validity, and Detection of Persons With Misfitting Responses." *Applied Psychological Measurement*, **34**, 410–423.

Rulison KL, Loken E (2009). "I've Fallen and I Can't Get Up: Can High-Ability Students Recover from Early Mistakes in CAT?" *Applied Psychological Measurement*, **33**, 83–101.

Stocking ML, Lewis C (1995). "A New Method of Controlling Item Exposure in Computerized Adaptive Testing." *Research Report No 95-25*, Educational Testing Service, Princeton, NJ.

Stocking ML, Lewis C (1998). "Controlling Item Exposure Conditional on Ability in Computerized Adaptive Testing." *Journal of Educational and Behavioral Statistics*, **23**, 57–75.

Swaminathan H, Gifford JA (1985). "Bayesian Estimation in the Two-Parameter Logistic Model." *Psychometrika*, **50**, 349–364.

Swaminathan H, Gifford JA (1986). "Bayesian Estimation in the Three-Parameter Logistic Model." *Psychometrika*, **51**, 589–601.

Urry VW (1970). *A Monte Carlo Investigation of Logistic Test Models*. Ph.D. thesis, Purdue University, West Lafayette, IN.

van der Linden WJ (1998). "Bayesian Item Selection Criteria for Adaptive Testing." *Psychometrika*, **63**, 201–216.

van der Linden WJ, Glas CAW (2000). *Computerized Adaptive Testing. Theory and Practice*. Kluwer, Boston.

van der Linden WJ, Pashley PJ (2000). "Item Selection and Ability Estimation in Adaptive Testing." In WJ van der Linden, CAW Glas (eds.), *Computerized Adaptive Testing. Theory and Practice*, pp. 1–25. Kluwer, Boston.

Veerkamp WJJ, Berger MPF (1997). "Some New Item Selection Criteria for Adaptive Testing." *Journal of Educational and Behavioral Statistics*, **22**, 203–226.

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

Wainer H (2010). "Fourteen Conversations about Three Things." *Journal of Educational and Behavioral Statistics*, **35**, 5–25.

Warm TA (1989). "Weighted Likelihood Estimation of Ability in Item Response Models." *Psychometrika*, **54**, 427–450.

Weiss D (1983). *New Horizons in Testing: Latent Trait Theory and Computerized Adaptive Testing*. Academic Press, New York.

Affiliation:

David Magis
Department of Mathematics (B37)
University of Liège
Grande Traverse 12
B-4000 Liège, Belgium
E-mail: david.magis@ulg.ac.be

Gilles Raîche
Département d'Education et Pédagogie
Université du Québec à Montréal
C.P. 8888, succursale Centre-Ville
Montréal (QC), Canada H3C 3P8
E-mail: raiche.gilles@uqam.ca

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 48, Issue 8 *Submitted:* 2011-04-22
May 2012 *Accepted:* 2012-01-09