ISSN 0005-1179, Automation and Remote Control, 2017, Vol. 78, No. 12, pp. 2165–2177. © Pleiades Publishing, Ltd., 2017.
Original Russian Text © A.I. Kibzun, E.A. Zharkov, 2017, published in Avtomatika i Telemekhanika, 2017, No. 12, pp. 84–99.

STOCHASTIC SYSTEMS

Two Algorithms for Estimating Test Complexity Levels

A. I. Kibzun[*,a] **and E. A. Zharkov**[*,b]

[*] *Moscow Aviation Institute, Moscow, Russia*
e-mail: [a] *kibzun@mail.ru,* [b] *zhr.evg@gmail.com*

Received December 7, 2016

Abstract—We study the problem of estimating the complexity levels of test problems and levels of preparation of the students that arises in learning management systems. To solve the problem, we propose two algorithms for processing test results. The first algorithm is based on the assumption that random answers of the test takers are described by a logistic distribution. To compute test problem complexities and levels of preparation of the students, we use the maximum likelihood method and the quasi-Newton Broyden–Fletcher–Goldfarb–Shanno optimization method, where the likelihood function is constructed in a special way based on Rasch's model. The second algorithm is heuristic and is based on recurrent recomputation of initial estimates obtained by adding up the positive answers of students separately by columns and rows of the matrix of answers, where columns correspond to answers of all students for a specific test, and rows correspond to answers of a specific student for all tests. We consider an example where we compare the results of applying the proposed algorithms.

Keywords: maximal likelihood method, recurrent recomputation, test complexity levels, levels of preparation of the students.

DOI: 10.1134/S0005117917120050

1. INTRODUCTION

Due to the development of learning management systems, there arises a need to automatically evaluate the knowledge of students based on a set of test problems they have done [1]. But all test problems have different levels of complexity, and students also have different levels of preparation unknown a priori. Therefore, the traditional approach to deriving a mark for every specific students by the number of correct answers to the problems cannot be considered objective. The approach for which each problem is expertly assigned a specific weight is also rather subjective. Therefore, a more objective estimate would be based only on the sample obtained from the answers of a group of students to the proposed test problems. But here there arises a problem of processing the resulting non-uniform sample.

The work [2] proposed an algorithm for getting an estimate of complexity levels of problems based on a single-parameter Rasch's model [3] and Newton's method for finding estimates that maximize the likelihood function. But the resulting algorithm had a narrow convergence region. The work [4] used to solve the same problem the quasi-Newton Broyden–Fletcher–Goldfarb–Shanno method proposed in [5], which had an infinite convergence radius, i.e., it did not present the problem of choosing the initial point. Besides, the work [4] established that the log-likelihood function is strictly concave with respect to parameters that define complexities of problems and preparation levels of students.

One drawback of the algorithm proposed in [4] was that it did not distinguish complexity levels for tests solved by the same number of students, even though the level of preparation of these students might be different.

In this work, we propose two test processing algorithms that do not have the drawback noted above. The first algorithm is based, similar to the algorithm from [4], on Rasch's model, the

maximum likelihood method, and quasi-Newton Broyden–Fletcher–Goldfarb–Shanno method. But the likelihood function here is different: we introduce to it special weight coefficients that let us differentiate problem complexities and preparation levels of students that have the same primary parameters. Besides, in this work we propose an exponential normalization of the estimates obtained by the first algorithm that take value on the entire real axis. This normalization let us reduce the grades to the commonly used 10-mark scale. The second algorithm is heuristic and is based on computing the complexity levels of test problems and preparation levels of students by adding up the columns and rows of the matrix of answers with subsequent recurrent recomputation of the resulting estimates by adding up the columns and rows of the matrix of answers with weights taken from the estimates of complexity levels of problems and preparation levels of students found on the previous step. Computations performed for numerous samples of original data indicate that estimates obtained with these two algorithms are virtually the same and do not contradict our intuition regarding the estimates. The nearly perfect matching of estimates obtained by two different algorithms implicitly validates that the algorithms work correctly.

2. PROBLEM SETTING

Consider the following model. Consider a group of I students who answer J test problems. Suppose that the result of solving problem j by student i is a random value X_{ij} with realizations from the set $\{0, 1\}$, i.e., we assume that $X_{ij} = 1$ if the problem was solved by student i and $X_{ij} = 0$ if not.

These random values can be represented as shown in Table 1.

Table 1. Model for answering test problems

	Problem 1	Problem 2	...	Problem J
Student 1	X_{11}	X_{12}	...	X_{1J}
Student 2	X_{21}	X_{22}	...	X_{2J}
...	...	...	...	...
Student I	X_{I1}	X_{I2}	...	X_{IJ}

Obviously, the correctness of answer X_{ij} depends both on the level δ_j of complexity of problem j and on the level θ_i of preparation of student i, i.e., $X_{ij} \triangleq X(\theta_i, \delta_j)$. Note that in this case in every row of Table 1 the preparation level of a student is determined by the same parameter θ_i, and in each column the problem's complexity is also constant and equal to δ_j. Suppose that all random values (RV) X_{ij}, $i = \overline{1, I}$, $j = \overline{1, J}$, are independent and have the same distribution defined by function

$$f(\theta_i, \delta_j) \triangleq \mathcal{P}\{X_{ij} = 1\} = 1 - \mathcal{P}\{X_{ij} = 0\}, \quad i = \overline{1, I}, \quad j = \overline{1, J}, \tag{1}$$

where function $f(\theta_i, \delta_j)$ takes values from 0 to 1 since it is the probability of the event $\{X_{ij} = 1\}$. Thus, random value X_{ij} has the distribution series shown in Table 2.

Table 2. Distribution series for RV X_{ij}

X_{ij}	0	1
$\mathcal{P}$	$1 - f(\theta_i, \delta_j)$	$f(\theta_i, \delta_j)$

As the mathematical model that relates the success of a student with his or her preparation level and problem complexity we use the Rasch's model [3]. Suppose that the distribution func-

tion $f(\theta_i, \delta_j)$ has logistic form

$$f(\theta_i, \delta_j) \triangleq \frac{\exp(b_j\theta_i - a_i\delta_j)}{1 + \exp(b_j\theta_i - a_i\delta_j)}, \quad i = \overline{1, I}, \quad j = \overline{1, J}, \tag{2}$$

where the weight

$$a_i \triangleq \frac{1}{J}\sum_{j=1}^{J} x_{ij} \tag{3}$$

characterizes the initial estimate for the preparation level of student i, and the weight

$$b_j \triangleq \frac{1}{I}\left(I - \sum_{i=1}^{I} x_{ij}\right) \tag{4}$$

characterizes the initial estimate for the complexity of problem j. Parameter θ_i of this function is called the preparation level of student i, $i = \overline{1, I}$; parameter δ_j, the complexity level of problem j, $j = \overline{1, J}$.

This logistic function is different from a similar function considered in [4] due to the presence of weight factors a_i and b_j. Note that the algorithm from [4] did not distinguish, for instance, the case when two different students solved one problem each, but one of these problems was solved by only one student in the group, while the second problem was solved by all students. It is clear that the preparation level of these two students is different, but the algorithm from [4] graded these students with exactly the same estimates. We will show below that introducing into the logistic function weight factors a_i and b_j lets us solve this problem. We also note that in the proposed mode, as the preparation level of a student increases from $-\infty$ to $+\infty$, the probability of his or her correct answer changes from zero to one, and as the complexity of a problem increases from $-\infty$ to $+\infty$, the probability of a correct answer reduces from one to zero.

Suppose that realizations x_{ij} of random values X_{ij} form a matrix $||x_{ij}||$, where $x_{ij} \in \{0, 1\}$, i is the index of a student in the group, $i = \overline{1, I}$, j is the index of the test problem, $j = \overline{1, J}$.

Remark 1. Suppose that the matrix of answers $||x_{ij}||$ does not have either rows or columns completely consisting of zeros or ones. We assume that if such columns appear in the matrix $||x_{ij}||$, we simply cross them out.

We formulate the following problem.

Problem. By a sample $\{x_{ij}\}$, estimate the complexity levels of test problems δ_j, $j = \overline{1, J}$, and preparation levels of students θ_i, $i = \overline{1, I}$.

3. ALGORITHM 1 BASED ON RASCH'S MODEL AND THE MAXIMAL LIKELIHOOD METHOD

We define Algorithm 1 for processing statistical data under the assumption that distribution function $f(\theta_i, \delta_j)$ has the form (2).

Consider the joint estimation procedure for unknown parameters θ_i, δ_j, $i = \overline{1, I}$, $j = \overline{1, J}$, based on the maximal likelihood method. Consider a matrix of observable answers $||x_{ij}||$, $i = \overline{1, I}$, $j = \overline{1, J}$. We exclude from the matrix all rows and columns that consist of exclusively either zeros or ones. Here, if $x_{ij} = 1$, $j = \overline{1, J}$ then we let $\overline{\theta}_i^* = 10$. If $x_{ij} = 0$, $j = \overline{1, J}$, we let $\overline{\theta}_i^* = 0$. Similarly, if $x_{ij} = 1$, $i = \overline{1, I}$, then $\overline{\delta}_j^* = 0$. And if $x_{ij} = 0$, $i = \overline{1, I}$, then $\overline{\delta}_j^* = 10$.

Using the explicit form of function $f(\theta_i, \delta_j)$ from (2), we can find the probability of incorrect answer by student i for test problem j:

$$\mathcal{P}\{X_{ij}(\theta_i, \delta_j) = 0\} = 1 - f(\theta_i, \delta_j) = 1 - \frac{\exp(b_j\theta_i - a_i\delta_j)}{1 + \exp(b_j\theta_i - a_i\delta_j)}$$
$$= \frac{1 + \exp(b_j\theta_i - a_i\delta_j) - \exp(b_j\theta_i - a_i\delta_j)}{1 + \exp(b_j\theta_i - a_i\delta_j)} = \frac{1}{1 + \exp(b_j\theta_i - a_i\delta_j)}.$$

Thus, probability of the fact that random value X_{ij} takes value $x_{ij} \in \{0, 1\}$ can be written as

$$\mathcal{P}\{X_{ij}(\theta_i, \delta_j) = x_{ij}\} = \frac{\exp(x_{ij}(b_j\theta_i - a_i\delta_j))}{1 + \exp(b_j\theta_i - a_i\delta_j)},$$

i.e., if $x_{ij} = 1$ then the numerator equals $\exp(b_j\theta_i - a_i\delta_j)$, and if $x_{ij} = 0$ then the numerator equals one.

We write the likelihood function

$$L(x, \theta, \delta) = \mathcal{P}\Big\{X_{ij}(\theta_i, \delta_j) = x_{ij}, i = \overline{1, I}, j = \overline{1, J}\Big\} = \frac{\prod\limits_{i=1}^{I}\prod\limits_{j=1}^{J}\exp(x_{ij}(b_j\theta_i - a_i\delta_j))}{\prod\limits_{i=1}^{I}\prod\limits_{j=1}^{J}(1 + \exp(b_j\theta_i - a_i\delta_j))},$$

where

$$x \triangleq \mathrm{col}(x_{11}, \ldots, x_{1J}, \ldots, x_{I1}, \ldots, x_{IJ}), \quad \theta \triangleq \mathrm{col}(\theta_1, \ldots, \theta_I), \quad \delta \triangleq \mathrm{col}(\delta_1, \ldots, \delta_J).$$

Then the log-likelihood function takes the form

$$\tilde{L}(x, \theta, \delta) \triangleq \ln L(x, \theta, \delta) = \sum_{i=1}^{I}\sum_{j=1}^{J}(b_j\theta_i - a_i\delta_j)x_{ij} - \sum_{i=1}^{I}\sum_{j=1}^{J}\ln(1 + \exp(b_j\theta_i - a_i\delta_j)). \tag{5}$$

We formulate the following problem:

$$(\theta^*, \delta^*) = \arg\max_{(\theta,\delta)} \tilde{L}(\theta, \delta). \tag{6}$$

Solving problem (6), one can estimate complexity levels for problems δ^* with respect to preparation levels of the students θ^*.

Remark 2. It is known [7] that estimates obtained based on the maximal likelihood approach are usually efficient, consistent, and asymptotically normal under minimal assumptions. This is a significant difference between them and other estimates, including estimates found by the least squares approach.

To solve problem (6), we study the properties of function (5); in particular, we prove that function (5) is strictly concave. We begin by reminding the definition of a strictly concave function.

Definition. Function $g(u) : \mathbb{R}^n \to \mathbb{R}^1$ is called strictly concave on a convex set $U \subset \mathbb{R}^n$ if for every $\lambda \in (0, 1)$ and arbitrary $u_1, u_2 \in U$ it holds that

$$g(\lambda u_1 + (1 - \lambda)u_2) > \lambda g(u_1) + (1 - \lambda)g(u_2).$$

We now formulate one of the main results of this work, which will imply that function (5) has a unique maximum.

Theorem 1. *Function $\tilde{L}(x,\theta,\delta)$ defined according to* (5) *is strictly concave with respect to θ and δ on $\mathbb{R}^I \times \mathbb{R}^J$.*

Proof of Theorem 1 is similar to the proof from [4] since the log-likelihood function in this case has the same linear structure as [4] and differs only by the presence of constant factors a_i and b_j.

Using the strict concavity of the log-likelihood function (5), we can find its maximum, which, as follows from the strict concavity of the log-likelihood function, is unique.

In order to maximize function (5) we use the quasi-Newton method of Broyden–Fletcher–Goldfarb–Shanno [5], which is based on accumulating information on the curvature of the objective function by observations on the changes in its gradient, which is a conceptual difference from Newton-like methods. The class of quasi-Newton methods excludes an explicit construction of the Hessian, replacing it with a certain approximation. Consider the following sequence constructed for some function $g(u) : \mathbb{R}^n \to \mathbb{R}^1$:

$$u_{k+1} = u_k + \alpha_k p_k, \tag{7}$$

where

$$p_k = -H_{k-1}^{-1} \nabla g(u_k),$$
$$H_{k+1} = H_k + \frac{y_k y_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k} - \frac{H_k s_k s_k^{\mathrm{T}} H_k}{s_k^{\mathrm{T}} H_k s_k},$$
$$s_k = u_{k+1} - u_k, \quad y_k = \nabla g(u_{k+1}) - \nabla g(u_k),$$

and α_k are found as a solution for the maximization problem

$$\alpha_k = \arg \max_{\alpha \in (0,1)} g\left(u_k + \alpha p_k\right).$$

If function $g(u_k + \alpha p_k)$ is unimodal with respect to scalar parameter α, then in order to solve the optimization problem we can use a simple dichotomy approach which converges very quickly on the $(0,1)$ interval.

Theorem 2 [5]. *Let H_0 be a symmetric negative definite matrix, let u_0 be the initial point, and suppose that the following conditions hold for some function $g(u) : \mathbb{R}^n \to \mathbb{R}^1$:*

(i) *function $g(u)$ is twice continuously differentiable;*

(ii) *the set $G \triangleq \{u \in \mathbb{R}^n : g(u) \geqslant f(u_0)\}$ is convex, and there exist negative constants c and C such that*

$$c \left\|z\right\|^2 \leqslant z^{\mathrm{T}} \nabla^2 g(u) z \leqslant C \left\|z\right\|^2$$

for all $z \in \mathbb{R}^n$ and $u \in G$.

Then sequence $\{u_k\}$, obtained according to (7)*, converges to the point u_*, which is the unique solution of problem*

$$u_* = \arg \max_{u \in G} g(u).$$

For convenience we introduce the notation $u \triangleq \mathrm{col}(\theta_1, \ldots, \theta_N, \delta_1, \ldots, \delta_L)$ and show that conditions of Theorem 2 hold for the function $g(u) = \tilde{L}(u)$.

Theorem 3. *Conditions of Theorem 2 hold for the function $g(u) \triangleq \tilde{L}(u)$, defined according to* (5).

Proof of Theorem 3 is similar to the proof shown in [4] since the log-likelihood function in this case has the same structure as [4] and differs only by the presence of constant factors.

Remark 3. We emphasize that procedure (7) in essence reduces the optimization problem in a multidimensional space to optimization of a scalar parameter $\alpha \in (0,1)$, which can be done by various optimization methods [8], for example by dichotomy.

As we have noted above, for Algorithm 1 constructed with the maximal likelihood method, the obtained estimates θ_i^* and δ_j^* can take values from $-\infty$ to $+\infty$. But here we establish the mutual location of estimates on the linear scale $(-\infty, +\infty)$. Therefore, we reduce, with exponential normalization, our estimates to a more commonly used 10-mark scale while preserving their mutual location.

For this purpose we let

$$\overline{\theta}_i^* = \begin{cases} 5\exp(d_1(\theta_i^* - \theta_m)), & \theta_i^* - \theta_m \leqslant 0 \\ 10 - 5\exp(d_2(\theta_i^* - \theta_m)), & \theta_i^* - \theta_m > 0, \end{cases}$$

where

$$\theta_m = \theta_{i_m}, \quad i_m = \arg\min_{i=\overline{1,I}} |a_i - 5|,$$

and d_1 and d_2 are some scaling parameters:

$$d_1 = \frac{\ln(\theta_{\min}/5)}{\theta_{\min}^* - \theta_m}, \quad d_2 = \frac{\ln((10-\theta_{\max})/5)}{\theta_{\max}^* - \theta_m},$$

where

$$\theta_{\min} = \frac{10}{J}\min_{i=\overline{1,I}}\sum_{j=1}^{J} x_{ij}, \quad \theta_{\max} = \frac{10}{J}\max_{i=\overline{1,I}}\sum_{j=1}^{J} x_{ij},$$
$$\theta_{\min}^* = \min_{i=\overline{1,I}} \theta_i^*, \quad \theta_{\max}^* = \max_{i=\overline{1,I}} \theta_i^*.$$

Thus, if $\theta_i^* = \theta_{\min}^*$ then $\overline{\theta}_i^* = \theta_{\min}$. If $\theta_i^* = \theta_{\max}^*$ then $\overline{\theta}_i^* = \theta_{\max}$.

The factor 5 was chosen so that the resulting estimates are symmetric with respect to θ_m, and the number θ_m itself is the average level of preparation that corresponds to the average level on the 10-mark scale, i.e., number 5.

Similarly, we let

$$\overline{\delta}_j^* = \begin{cases} 5\exp(c_1(\delta_j^* - \delta_m)), & \delta_j^* - \delta_m \leqslant 0 \\ 10 - 5\exp(c_2(\delta_j^* - \delta_m)), & \delta_j^* - \delta_m > 0, \end{cases}$$

where

$$\delta_m = \delta_{j_m}, \quad J_m = \arg\min_{j=\overline{1,J}} |b_j - 5|,$$
$$c_1 = \frac{\ln(\delta_{\min}/5)}{\delta_{\min}^* - \delta_m}, \quad c_2 = \frac{\ln((10-\delta_{\max})/5)}{\delta_{\max}^* - \delta_m},$$
$$\delta_{\min} = \frac{10}{I}\min_{j=\overline{1,J}}\left(I - \sum_{i=1}^{I} x_{ij}\right), \quad \delta_{\max} = \frac{10}{I}\max_{j=\overline{1,J}}\left(I - \sum_{i=1}^{I} x_{ij}\right),$$
$$\delta_{\min}^* = \min_{j=\overline{1,J}} \delta_j^*, \quad \delta_{\max}^* = \max_{j=\overline{1,J}} \delta_j^*.$$

In this case, if $\delta_j^* = \delta_{\min}^*$ then $\overline{\delta}_i^* = \delta_{\min}$. If $\delta_i^* = \delta_{\max}^*$, then $\overline{\delta}_i^* = \delta_{\max}$.

4. ALGORITHM 2 BASED ON RECURRENT RECOMPUTATION OF COMPLEXITY LEVELS OF PROBLEMS AND PREPARATION LEVELS OF STUDENTS

We define Algorithm 2 for solving the above problem, which is in essence a heuristic algorithm based on the idea that the traditional way of estimation, when the grade of a student is based on the number of problems he or she has solved correctly, must be corrected with regard to the complexity of solved problems. But the complexity levels of problems are not known in advance. Therefore, in Algorithm 2 proposed below we recurrently refine the preparation levels of students and complexity levels of test problems. Consider a matrix $||x_{ij}||$ consisting of zeros and ones with elements

$$x_{ij} \in \{0,1\}, \quad i = \overline{1,I}, \quad j = \overline{1,J}.$$

Suppose that $I \geqslant 5$, $J \geqslant 5$, since the case of a small sample is of no practical interest. First, similar to Section 3, we exclude from the matrix $||x_{ij}||$ columns and rows that contain either all ones or all zeros. Suppose that the student's mark $\theta_i = 10$ if the ith row contains all ones and $\theta_i = 0$ if it contains all zeros. Similarly, we let $\delta_j = 10$ if column j contains all zeros and $\delta_j = 0$ if that column contains all ones. Next we process the rest of the elements of matrix $||x_{ij}||$, assuming without loss of generality that there are I rows and J columns left.

We add up the columns and rows of the matrix separately and get numbers that correspond to the number of students who have solved problem j and the number of problems solved by student i:

$$a_i^0 = \sum_{j=1}^{J} x_{ij}, \quad b_j^0 = \sum_{i=1}^{I} x_{ij}, \quad i = \overline{1,I}, \quad j = \overline{1,J}.$$

Next we compute normalized values of the resulting sums that give us initial characteristics for the preparation level of student i and complexity level of problem j corresponding to the 10-mark scale:

$$\overline{a}_i^0 = \frac{10}{J} a_i^0, \quad \overline{b}_j^0 = \frac{10}{I}(I - b_j^0), \quad i = \overline{1,I}, \quad j = \overline{1,J}. \tag{8}$$

Next we compute new sums of columns and rows but with the resulting weights:

$$a_i^1 = \sum_{j=1}^{J} \overline{b}_j^0 x_{ij}, \quad i = \overline{1,I},$$

$$b_j^1 = \sum_{i=1}^{I} \overline{a}_i^0 x_{ij}, \quad j = \overline{1,J}.$$

Next we compute new values for the preparation level of student i and complexity level of problem j:

$$\overline{a}_i^1 = a_i^1 \frac{10}{\sum\limits_{j=1}^{J} \overline{b}_j^0}, \quad i = \overline{1,I}, \tag{9}$$

$$\overline{b}_j^1 = \left(\sum_{i=1}^{I} \overline{a}_i^0 - b_j^1 \right) \frac{10}{\sum\limits_{i=1}^{I} \overline{a}_i^0}, \quad j = \overline{1,J}. \tag{10}$$

We continue the iterative process, computing $\overline{a}_i^k$ and $\overline{b}_j^k$, $k = 2, 3, \ldots$. Suppose that these sequences converge:

$$\overline{a}_i^k \to a_i^*, \quad \overline{b}_j^k \to b_j^*, \quad i = \overline{1,I}, \quad j = \overline{1,J}.$$

First note that these limit values, if they exist, cannot all be equal to zero at the same time. We assume, for example, that $\bar{b}_j^k \to 0$. Then according to (9)

$$\bar{a}_i^k \to \sum_{j=1}^{J} > \Delta_i > 0,$$

since by assumption all x_{ij}, $j = \overline{1,J}$ cannot be equal to zero. Therefore, according to (10): $\bar{b}_j^k \to b_j^* > 0$, since by assumption all x_{ij}, $i = \overline{1,I}$, cannot be equal to one at the same time. We have arrived at a contradiction.

The resulting limit values, if they exist, will be interpreted as estimates of the preparation level of student i and complexity level of problem j:

$$\hat{\theta}_i^* \triangleq a_i^*, \quad i = \overline{1,I}, \quad \hat{\delta}_j^* \triangleq b_j^*, \quad j = \overline{1,J}. \tag{11}$$

Let us obtain necessary conditions for the convergence of Algorithm 2. With this purpose we let

$$a_i \stackrel{\text{def}}{=} \frac{\overline{a}_i^k}{10} = \frac{\overline{a}_i^{k+1}}{10} \quad \text{and} \quad b_j \stackrel{\text{def}}{=} \frac{\overline{b}_j^k}{10} = \frac{\overline{b}_j^{k+1}}{10}$$

and according to (9) and (10) we get the following relations:

$$a_i = \frac{\sum\limits_{k=1}^{J} b_k x_{ik}}{\sum\limits_{k=1}^{J} b_k}, \quad i = \overline{1,I}, \tag{12}$$

$$b_j = \frac{\sum\limits_{l=1}^{I} a_l - \sum\limits_{l=1}^{I} a_l x_{lj}}{\sum\limits_{l=1}^{I} a_l}, \quad j = \overline{1,J}. \tag{13}$$

Thus, we get that limit values a_i, $i = \overline{1,I}$, and b_j, $j = \overline{1,J}$, if they exist, must satisfy the following system of nonlinear equations:

$$\sum_{k=1}^{J} (a_i - x_{ik}) b_k = 0, \quad i = \overline{1,I}, \tag{14}$$

$$\sum_{l=1}^{I} (x_{lj} - 1 + b_j) a_l = 0, \quad j = \overline{1,J}, \tag{15}$$

excluding the case $a_i = 0$, $i = \overline{1,I}$ and $b_j = 0$, $j = \overline{1,J}$, since in this case uncertainty arises in Eqs. (12) and (13) and in every row and every column of matrix $||x_{ij}||$ not all elements are equal to zero or one simultaneously. Substituting one equation into another and vice versa, we get two independent systems of quadratic equations:

$$\sum_{l=1}^{I} (x_{lj} - 1 + b_j) \sum_{k=1}^{J} b_k x_{lk} = 0, \quad j = \overline{1,J}, \tag{16}$$

$$\sum_{k=1}^{J} (a_i - x_{ik}) \sum_{l=1}^{I} a_l (1 - x_{lk}) = 0, \quad i = \overline{1,I}. \tag{17}$$

Let us show that the resulting system has only one admissible solution $a_i \in (0,1)$, $i = \overline{1,I}$, $b_j \in (0,1)$, $j = \overline{1,J}$. Note that each equation has only a_i^2, b_j^2 appearing squared. For instance, consider Eq. (16) for $j = 1$. We conclude that the equation has the following structure of a quadratic equation:

$$cb_1^2 + db_1 + p = 0,$$

where

$$c = \sum_{l=1}^{I} x_{l1}, \quad d = \sum_{k=2}^{J} b_k \sum_{l=1}^{I} x_{lk}, \quad p = \sum_{k=2}^{J} b_k \left(\sum_{l=1}^{I} x_{lk}(x_{l1} - 1) \right).$$

Note that the discriminant of this equation equals $D = d^2 - 4pc$. But the free coefficient $c < 0$ since all b_k are positive according to (13), factors x_{lk} are equal to zero or one, and by assumption all x_{l1} cannot be equal to one at the same time. Therefore, $\sqrt{D} > d$, and $d > 0$. Consequently, the quadratic equation in question can have only one positive root

$$b_1(b_2, \ldots, b_J) = \frac{1}{2c}(-d + \sqrt{D}),$$

which according to (13) satisfies condition $0 < b_1 < 1$, i.e. is admissible. Note that function $b_1(b_2, \ldots, b_J)$ is convex and strictly increasing with respect to every variable. Similarly, one can show that Eqs. (16) for other $j = \overline{2,J}$ have only one admissible root b_j. If we now fix b_j in every equation numbered $j = \overline{1,J}$, then the system of Eqs. (16) becomes linear with respect to variables $b_1, \ldots, b_J$, and it has a nonzero determinant since b_j^2 occur only in their own equation with index j. Therefore, the resulting system of linear equations is feasible and has a unique admissible solution. The same considerations could be done for the system of Eqs. (17), showing that it has only one admissible solution.

Remark 4. Note that by choosing the initial approximation we can influence the convergence rate of the iterative process. In essence, the proposed iterative algorithm lets us refine the initial approximation, which is chosen in such a way as one traditionally grades a student: how many problems he or she has solved out of the 10 available, that is the grade. The iterative algorithm corrects this grade by taking into account problem complexities. In turn, test complexity levels are refined according to the information on how the students were solving test problems.

Remark 5. Unfortunately, we cannot prove that estimates, obtained with the proposed Algorithm 2 converge to "correct" values. But numerous computations done for different initial data and under large dimensions indicate that the normalized estimates obtained with Algorithm 2 virtually coincide with normalized estimates resulting from Algorithm 1 based on the maximum likelihood method. In particular, this effect is demonstrated in the numerical example that we show below. Note also that the problem dimension must be sufficiently large, with $I > 4$ and $J > 4$, which we observe in practical problems. For small dimensions, e.g., for $I = 2$ and $J = 3$, one can construct a counterexample where Algorithm 2 enters a loop. This effect is probably due to the lack of averaging for small samples; there is no looping on large samples. Therefore,, it appears that sufficient conditions for the convergence of Algorithm 2 is related to the problem dimension, but the study of this question falls outside the scope of this paper.

Let us show the results of processing test results on a 10-mark scale, assuming that

$$\overline{\theta}^*_{\max} = \hat{\theta}_{\max}, \quad \overline{\theta}^*_{\min} = \hat{\theta}_{\min}, \quad \overline{\delta}^*_{\max} = \hat{\theta}_{\max}, \quad \overline{\delta}^*_{\min} = \hat{\theta}_{\min},$$

where

$$\hat{\theta}_{\max} = \max_{i=\overline{1,I}} \frac{10}{J} \sum_{j=1}^{J} x_{ij}, \quad \hat{\theta}_{\min} = \min_{i=\overline{1,I}} \frac{10}{J} \sum_{j=1}^{J} x_{ij},$$
$$\hat{\delta}_{\max} = \max_{j=\overline{1,J}} \frac{10}{I} \left(I - \sum_{i=1}^{I} x_{ij} \right), \quad \hat{\delta}_{\min} = \min_{j=\overline{1,J}} \frac{10}{I} \left(I - \sum_{i=1}^{I} x_{ij} \right).$$

Besides, we find in the resulting solution (θ^*, δ^*) maximal and minimal values:

$$\theta^*_{\max} = \max_{i=\overline{1,I}} \theta^*_i, \quad i_{\max} = \arg \max_{i=\overline{1,I}} \theta^*_i,$$
$$\theta^*_{\min} = \min_{i=\overline{1,I}} \theta^*_i, \quad i_{\min} = \arg \min_{i=\overline{1,I}} \theta^*_i,$$
$$\delta^*_{\max} = \max_{j=\overline{1,J}} \delta^*_j, \quad j_{\max} = \arg \max_{j=\overline{1,J}} \delta^*_j,$$
$$\delta^*_{\min} = \max_{j=\overline{1,J}} \delta^*_j, \quad j_{\min} = \arg \max_{j=\overline{1,J}} \delta^*_j.$$

Converting values (θ^*, δ^*) to a 10-mark scale, we let

$$\overline{\theta}^*_{i_{\max}} = \hat{\theta}_{\max}, \quad \overline{\theta}^*_{i_{\min}} = \hat{\theta}_{\min},$$
$$\overline{\delta}^*_{j_{\max}} = \hat{\delta}_{\max}, \quad \overline{\delta}^*_{j_{\min}} = \hat{\delta}_{\min}.$$

Next we find the recomputation coefficients in relations

$$\overline{\theta}^*_i = a\theta^*_i + b, \quad \overline{\delta}^*_j = c\delta^*_j + d,$$

using equations

$$a\theta^*_{i_{\max}} + b = \hat{\theta}_{\max}, \quad a\theta^*_{i_{\min}} + b = \hat{\theta}_{\min},$$
$$c\delta^*_{j_{\max}} + d = \hat{\delta}_{\max}, \quad c\delta^*_{j_{\min}} + d = \hat{\delta}_{\min}.$$

Then we get that

$$a = \frac{\hat{\theta}_{\max} - \hat{\theta}_{\min}}{\theta^*_{i_{\max}} - \theta^*_{i_{\min}}}, \quad b = \hat{\theta}_{\max} - a\theta^*_{i_{\max}},$$
$$c = \frac{\hat{\delta}_{\max} - \hat{\delta}_{\min}}{\delta^*_{j_{\max}} - \delta^*_{j_{\min}}}, \quad d = \hat{\delta}_{\max} - c\delta^*_{j_{\max}}.$$

Finally, to these results we ca add estimates for those students and problems that initially obtained maximal and minimal marks equal to 10 and 0.

5. NUMERICAL EXAMPLE

As a sample problem to illustrate the efficient operation of the proposed algorithm for estimating complexity levels of problems, we consider a simple example.

Table 3 shows the results of students passing the tests, picked in such a way as to observe both very weak and very strong students.

Note that number $a^0_i = a_i$ corresponds to the grade obtained by student i via traditional grading on a 10-mark scale, and $b^0_j = b_j$ corresponds to a traditional estimate of the complexity level for problem j.

Table 3. Results of solving test problems $||x_{ij}||$

Problem j / Student i	1	2	3	4	5	$a_i^0 = \frac{10}{J}\sum_{j=1}^{J} x_{ij}$
1	1	0	0	0	0	2
2	0	0	0	0	1	2
3	0	1	0	1	1	6
4	0	1	1	1	1	8
5	0	1	1	1	1	8
$b_j^0 = \frac{10}{I}\left(I - \sum_{i=1}^{I} x_{ij}\right)$	8	4	6	4	2	

Table 3 shows that students 1 and 2 solved only one problem each, but student 1 solved a problem that nobody else had solved, while the second solved only the last problem that was also solved by all other students.

Consider the operation of Algorithm 2 based on recurrent recomputation of complexity levels of problems and preparation levels of students. The solution is a vector of problem complexities equal to

$$\delta_2^* = \operatorname{col}\Big(7.99,\ \ 2.52,\ \ 4.29,\ \ 2.52,\ \ 2.01\Big),$$

and vector of student preparation levels equal to

$$\theta_2^* = \operatorname{col}\Big(4.13,\ \ 1.04,\ \ 3.65,\ \ 5.86,\ \ 5.86\Big).$$

We give resulting values after a linear normalization to the 10-mark scale:

$$\overline{\delta}_2^* = \operatorname{col}\Big(8.0,\ \ 2.51,\ \ 4.29,\ \ 2.51,\ \ 2.0\Big),$$
$$\overline{\theta}_2^* = \operatorname{col}\Big(5.84,\ \ 2.0,\ \ 5.24,\ \ 8.0,\ \ 8.0\Big).$$

Here Algorithm 2 needed 13 iterations for the process to converge up to the second decimal place.

Table 3 shows that problems 2 and 4 were solved by the same students. Therefore, complexity levels of these problems (2.51) are the same. Problem 5 was solved by almost all students except student 1. Therefore, complexity level of this problem is the lowest, 2.0. At the same time, only one student solved problem 1, therefore, the complexity level of this problem is the highest, 8.0.

Although student 1 solved only one problem, but the hardest one, her preparation level is estimated by value 5.84. At the same time, preparation level for student 2 is the lowest, equal to 2.0, since he solved only the easiest problem. Preparation levels of students 4 and 5 are the highest since they solved almost all problems, except for 1, and the level of these students turned out to be higher than for student 1 who solved exactly the one problem that students 4 and 5 did not; but student 1 solved only one problem, even though it was the hardest one.

Let us now consider the results of the operation of an algorithm based on the maximum likelihood method with respect to proposed weight coefficients.

As the starting point for the Broyden–Fletcher–Goldfarb–Shanno method, we use the zero vector

$$u_0 = \operatorname{col}\Big(0,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0\Big).$$

The vector of problem complexities in this case turned out to be equal to

$$\delta_1^* = \operatorname{col}\Big(5.06,\ \ -4.11,\ \ -0.53,\ \ -4.11,\ \ -6.08\Big).$$

Table 4. Preparation levels of students $\theta_i^*,\ i=\overline{1,5}$

Algorithm 1	Exponential normalization	5.58	2.0	5.0	8.0	8.0
Algorithm 2	Linear normalization	5.84	2.0	5.24	8.0	8.0

Table 5. Complexity level of problem $\delta_j^*,\ j=\overline{1,5}$

Algorithm 1	Exponential normalization	8.0	2.77	5.0	2.77	2.0
Algorithm 2	Linear normalization	8.0	2.51	4.29	2.51	2.0

The vector of preparation levels of students turned out to be equal to

$$\theta_1^* = \operatorname{col}\Big(-1.75,\ \ -6.98,\ \ -2.40,\ \ 2.47,\ \ 2.47\Big).$$

We give the resulting estimates in the 10-mark scale with the exponential normalization defined above:

$$\overline{\delta}_1^* = \operatorname{col}\Big(8.0,\ \ 2.77,\ \ 5.0,\ \ 2.77,\ \ 2.0\Big),$$
$$\overline{\theta}_1^* = \operatorname{col}\Big(5.58,\ \ 2.0,\ \ 5.0,\ \ 8.0,\ \ 8.0\Big).$$

Tables 4 and 5 show results that indicate that the introduced weights let us differentiate students who solved only one problem each, but with different complexity. Results of applying Algorithm 2 are very similar to the results of applying Algorithm 1. The relative position of estimates for student preparation and complexity levels of problems in both algorithms is the same, and they look more objective than estimates obtained based on the algorithm from [4]. Note that a change in the weights of the terms in Algorithm 1 with respect to proposed values leads to completely different results that do not correspond to the results of Algorithm 2 and intuitive considerations.

We have chosen a simple example so that it would be easy to check the results of the resulting estimates with intuitive considerations. In high-dimensional samples it would be hard to perform this kind of analysis.

The operation of the algorithms has been tested on examples of high dimension, in particular for a 100×100 matrix, i.e., for 100 students and 100 problems. Here the numerous initial data were generated randomly, since a full enumeration would contain an astronomical number of possibilities. Note that both algorithms in all cases gave close estimates, which implicitly confirms that Algorithm 2 works correctly, and the iterative procedure converges to "correct" estimates close to the estimates obtained with Algorithm 1. Here the total time of operation for Algorithm 2 was 0.19 seconds. Note that the operation time of Algorithm 2 increases insignificantly when problem dimension grows, since, first, it has relatively low algorithmic complexity, and second, as the dimension grows the number of iterations needed for convergence remains virtually unchanged. The total running time of Algorithm 1 was 2.8 seconds, which is naturally due to the algorithmic complexity and high number of iterations. We performed computations on an Intel Core i5-6600 3500 MHz CPU and 8 GB of RAM. To speed up computation, we have used the Cython extension and the Numpy/SciPy libraries.

Remark 6. Note that numerous computations for different initial data indicate that heuristic Algorithm 2 yields approximately the same estimates as Algorithm 1 based on the maximal likelihood approach. Stability of Algorithm 2 seems to be related to the law of large numbers, when on every iteration we add small corrections to current estimates with different signs, and the number of corrections grows as iterations accumulate. But this effect requires additional study.

Remark 7. We emphasize that the resulting test complexity levels have been found not with expert estimates of the teachers but based on the results shown by the students on these tests.

Remark 8. These algorithms are used in the learning management system CLASS.NET [6] used in Moscow Aviation Institute to study mathematical disciplines.

6. CONCLUSION

In this work, we consider the problem of constructing estimates for complexity levels of problems and preparation levels of students based on their results shown on test problems.

To solve this problem we have proposed two algorithms for test processing. The first algorithm is based on Rasch's model, the maximal likelihood method, and the quasi-Newton method of Broyden–Fletcher–Goldfarb–Shanno for maximizing the likelihood function. The second algorithm is heuristic and is based on recurrently refining the estimates obtained by adding up the rows and columns of the matrix of answers with weights found on the previous step.

Both algorithms successfully process all matrices of answers, yielding close results, and also correctly distinguish the complexities of problems and preparation levels of students that have the same initial parameters. However, Algorithm 2 proves to be more efficient in its operation than Algorithm 1 since it has low algorithmic complexity.

The main result of this work is the detection of an effect that heuristic Algorithm 2 yields approximately the same estimates as Algorithm 1 based on maximizing the likelihood function for the logistic Rasch's model.

Efficiency of the proposed methods has been tested with statistical data on student results in a learning management system CLASS.NET.

ACKNOWLEDGMENTS

This work was done as part of fulfilling the Government contract by the Russian Ministry of Science and Education no. 2.2461.2017/PCh.

REFERENCES

1. Kibzun, A.I., Karolinskaya, S.N., and Shayukov, R.I., A Remote Study System for Mathematical Disciplines in the University, *Vestn. Komp'yut. Inform. Tekhnol.*, 2006, no. 4, pp. 29–36.
2. Kibzun, A.I. and Panarin, S.I., Generation of Integral Rating by Statistical Processing of the Test Results, *Autom. Remote Control*, 2012, vol. 73, no. 6. pp. 1029–1045.
3. Rasch, G., *Probabilistic Models for Some Intelligence and Attainment Tests*, Chicago: Univ. of Chicago Press, 1980.
4. Kibzun, A.I. and Inozemtsev, A.O., Using the Maximum Likelihood Method to Estimate Test Complexity Levels, *Autom. Remote Control*, 2014, vol. 75, no. 4, pp. 607–621.
5. Nocedal, J. and Wright, S.J., *Numerical Optimization*, New York: Springer, 2006, 2nd ed.
6. Kibzun, A.I., Vishnyakov, B.V., and Panarin, S.I., Interface of a Remote Study System for Mathematical Disciplines, *Vestn. Komp'yut. Inform. Tekhnol.*, 2008, no. 10, pp. 43–48.
7. Ivchenko, G.I. and Medvedev, Yu.I., *Matematicheskaya statistika* (Mathematical Statistics), Moscow: Vysshaya Shkola, 1984.
8. Polyak, B.T., *Vvedenie v optimizatsiyu* (Introduction to Optimization), Moscow: Nauka, 1983.

This paper was recommended for publication by P.V. Pakshin, a member of the Editorial Board