A Novel Student Ranking Algorithm for Off-line Examinations

Vinayak G. Padmaji, Prasanna Chaporkar, Madhu N. Belur
Department of Electrical Engineering
Indian Institute of Technology Bombay, India

Abstract

Computerization of the examination process has facilitated fast and unbiased large-scale evaluation, by using the popular format of multiple choice based question papers. The primary drawback of this format is the possibility of guessing by the students. This has been largely remedied by online Computerized Adaptive Testing (CAT) methods where the next question presented to a student depends upon his/her responses to earlier questions of known difficulty levels. The student capabilities are evaluated in real time. However, the case where the questions' difficulty levels are not apriori known, but in fact estimated aposteriori with respect to aggregate student performance, has been less analyzed. This paper addresses the problem of off-line estimation of the student capability levels using a maximum likelihood estimation method and proposes to use these estimated capability levels rather than raw-marks for the purpose of ranking of students. We focus on the case when guessing is ruled out, for example, due to exact numerical answer possibilities instead of multiple choices. With respect to suitable performance objectives, like low errors in rank-allotment, the estimated-capability based ranking emerges more reliable than the traditional marks based ranking of students.

1. Introduction

Often the purpose of a large scale examination is to select students most suitable to be admitted to reputed institutions. These examinations have evolved a lot in the way they are conducted as the number of students taking these exploded. The evaluation of large number of students answering subjective questions is a very difficult problem to tackle logistically. Hence, computerization of examination conduction and evaluation are being used. The format of examination which leverages the computational capability of current machines is the multiple choice based format and is currently the most popular way to select the eligible students. In such a test each question has a certain number of options, usually 4 or 5, one of which is correct and to get credit the student has to mark the correct option by clicking on the right answer. The key problem faced when using multiple choice based tests is the guessing possibility. For example, if each question has four choices, then a student who randomly chooses answers to all questions may easily get a quarter of the total marks and much more if he/she gets lucky. These marks obtained by pure guess work may turn out to be greater than the marks obtained by a sincere student who does not guess. Therefore, multiple choice questions with guessing possibility creates a scenario where students with lower capabilities could be ranked better than students who are more capable.

Many solutions have been suggested to counter the guessing issue; mainly by the use of negative marking. Many scoring schemes have been studied and their effect on the reliability of tests have been investigated. The schemes studied vary from penalizing wrongly answered questions to awarding marks for unanswered questions in order to discourage guessing. A survey of these scoring methods can be found in [1]. See also [2, 3] for a thorough analysis of different negative marking schemes and the (marginal) probabilities of getting marks. If a student answers a large number of questions then one gets a good approximation of the student's capability level. However, this increases the duration of the examination, which is undesirable due to requirement of more resources for conducting the examination, especially in the case of large scale examinations.

The most successful among the evaluation methods has been the Online Computerized Adaptive Testing methods. Computerized Adaptive Testing (CAT) is a form of computer based test which adapts to the student performance on questions in real time. Central to the CAT methodology is the large question repository which contains pre-calibrated questions, i.e. their "difficulty" levels are known apriori. The challenge is to choose the difficulty level of the next question to be given to a student based on his/her performance in the previous questions of the exam. Once the difficulty level is determined, an appropriate question from the pre-calibrated question repository is chosen. The objective is to determine the student's "capability" within a desired confidence interval. CAT is designed to achieve this

goal with lesser number of questions, thus saving on redundancy and time. During CAT, new questions, i.e. questions whose difficulty levels are yet to be determined, are fielded to the students and the capability levels of the students, together with the students' performance on the new question, play a role in deciding the new questions' difficulty levels. For example, in GRE this is done by having the students answer a section within the test which does not add to their actual scoring for the test. Therefore, intrinsic to this adaptive testing is a clear separation of the questions:

- Either question has been pre-calibrated to a difficulty level, and hence the question is used to assign the student's capability level, or
- The question is being calibrated, and hence the student's capability level is used to assign the question's difficulty level.

Of course, it is the aggregate performance that decides the precise value of the assigned level; a confidence interval is also simultaneously calculated. The aggregate aspect systematically filters out noise in the data due to guessing possibilities by students. An attractive feature of CAT is that a question is termed more difficult if less number of students have answered it correctly, and a student is termed more capable if harder questions have been answered correctly. A thorough description of CAT and item response theory can be found in [4, 5, 6].

Although CAT seems to be the solution for the computerized evaluation problem, it has a few key limitations which make it a less attractive alternative. These are listed below:

- It requires a huge repository of calibrated questions from which questions have to be picked. Creating questions with cleverly placed choices and the calibration of these questions by having large number of students answer them, both require great of amount time to execute.
- The students who have completed the test are required to be secretive about the questions they answered, but that hardly ever happens. Over time a difficult question therefore becomes a easy question as everyone comes to know about it. Thus, CAT also requires that the questions used from the repository be re-calibrated periodically. This is not a feasible idea when there are more than hundred thousand students, taking the exam simultaneously or in quick succession, bankrupting the repository very quickly and leading to the recalibration of many questions every time.
- Huge amount of resources are required to implement the CAT system such as computers, proprietary software and networking issues, security from hackers who may try to access the question repository.

CAT is able to mitigate the impact of guessing by finding the most likely capability using the least number of questions, but it cannot completely eliminate the impact of guessing. Most exams which use this methodology, provide a score to each student and an eligibility criterion to the institutions to admit or reject a student's application for admission. This is not useful when students are to be admitted to prestigious institutions based on ranking obtained in the exam. CAT, though powerful and proven (mostly) to be accurate seems to be unfeasible for large scale examinations, with prohibitive infrastructure requirements, that needs to be conducted in a short duration.

In this paper, we consider an off-line examination as opposed to the on-line adaptive examinations. An off-line examination has the following features:

- The difficulty levels of the questions being presented to the students are unknown. As a consequence, there is no incentive to assign different weightages to different questions and all the questions fetch the same number of marks.
- Student responses are not evaluated in real-time. The evaluation of a student's response to each question takes place separately after the examination.
- Marks scored by a student are considered equivalent to student capability. Higher the marks scored, more capable the student.

Thus unlike in adaptive testing, it is difficult to mitigate the impact of guessing on the student ranking in an off-line examination. Negative marking evaluation schemes are not effective [3]. Hence, a good possibility can be removing guessing possibility altogether. This is done by introducing question formats which are devoid of choices and require students to type in numerical answers into answer boxes.

A concrete example of an examination implementing the above features is the important *Graduate Aptitude Test in Engineering* (GATE), which is conducted annually in India. Many of the instances of this exam allow exact numerical answer-format, instead of the traditional multiple-choice format: see [7] for more details of this exam. Moreover, a very small fraction of students are typically considered for admission to reputed Indian institutions. This exam has had a registration count of more than a *million* students in the recent years and is conducted over 2 weekends, across various examination centers in the country. An eligibility criteria based on scores is decided upon and usually 15% of students qualify. For further admission procedures, ranks of the students are used. Each student is allotted a rank which mirrors the student's capability and the universities admit students based

on the rank. Given the magnitude and impact of this examination on students and institutions, it is very crucial to have a reliable assignment of ranks to students in off-line examinations.

This paper addresses this central question with respect to an off-line examination: "Given that guessing is not allowed, are marks scored by the students an exact representation of their capabilities?" This is an important question to be answered because student behavior in an examination hall is a complex interplay between factors such as intelligence, preparedness, risk taking nature and instructions given before the examination; see [1]. We assume that capability levels are mainly dependent upon the intelligence and preparedness. The capability value of a student is a convex combination of these two factors. In order to make the distinction between these clear, some students who have memorized formulae (well prepared) with no real understanding of a difficult concept answer quickly, giving them more time to answer other questions. Whereas a student who has understood the difficult concept (more intelligent) might invest time in deriving the equations needed to solve the question and end up losing time. There is also a chance that, no matter how well-prepared and intelligent a student is, one could commit a simple mistake and end up losing some marks. These factors affect the way a student performs and should be considered while evaluation and rank allocation.

We illustrate the impact of presence or absence of guessing on student ranking with a simple example. Consider students A,B and C answering a test which has 5 questions. The difficulty levels of the questions are known a priori and the students are instructed not skip any questions. Suppose students A, B and C obtained 2, 2 and 3 marks respectively. Moreover, student A has answered 2 difficult questions correctly. Student B has answered 2 easy questions correctly. Student C has answered 2 easy questions and 1 difficult question correctly. If guessing was allowed, then it is most likely that student A, who was unable to answer easy questions, should have guessed the difficult questions and will be penalized. Student B would be considered more capable than student A. Student C is considered the most capable as answering both types of questions increased his credibility. Therefore we may conclude that C > B > A is more likely. On the other hand, if guessing was not allowed and ranking is marks based then we can conclude that C > B = A. Both of these conclusions are incorrect, as its quite straight forward that most likely ranking should be A > C > B. This illustration highlights 2 major points:

1. Ranking process is very different when guessing is not allowed than that with guessing.
2. Ranking based on raw score may not be close to the actual ranking.

An important observation to be noted is that we could make conclusions about ranking the students easily because the difficulty levels of questions were known. When difficulty levels of questions are unknown, it is even more difficult to rank students. Moreover, in the second case, we see that students B and A have got the same marks and we have a tie. It would be difficult to break such a tie when difficulty levels are not known. Thus, difficulty levels play a important role in assigning student capabilities and ranking.

To the best of our knowledge, the problem of ranking students in the case of an off-line examination has not been addressed. In this paper, we take the first step in this direction. Specifically, we consider ranking in off-line examinations with the features described earlier. In this scenario, we propose a novel maximum likelihood algorithm to determine student capabilities and thereby their ranks. Using simulation studies we show that ranking based on student capabilities performs better than marks based ranking.

The paper is organized as follows. Section 2 contains various definitions of parameters used in the paper. Section 3 describes the proposed iterative algorithm for off-line determination of student-capability/question-difficulty levels. Section 4 contains the description and results of various simulation and numerical experiments we conducted in the FOSS numerical computation package Scilab [8]. Section 5 contains some concluding remarks and future work.

2. Definitions and Performance metrics

We define the terms we use frequently in the paper: these definitions have undergone significant changes compared to our related work in [9].

- **Capability level**: Capability level can be thought of as a function of two main factors viz., the IQ and the preparedness of the student (more details in Section 1). The most eligible student is the one who has both greater IQ and has prepared well for the exam. The student capability level values lie in the range [0,1]. 0 and 1 capability levels correspond to the least capable and the most capable student respectively. A student who is more capable has a higher probability of a getting a correct answer given any question.
- **Difficulty level**: Difficulty level is a value assigned to a question which lies in the range [0,1]. Values closer to zero identify it as a easier question and values closer to one denote a difficult question. If a question is more difficult, then any student will have a lower probability of answering it correctly. Note that in this paper, we use difficulty level for

questions in a complementary way from the easiness level definition used in [9].

- **Confidence level**: For a given question (of a certain difficulty level, to be defined next) and for a given student (of a certain capability level), we assign a 'confidence level' as the probability with which that student answers the given question correctly. It is equal to the proportion of correct responses by a student if he/she answers a large number of questions with the same difficulty level.
- **Rank**: The rank of a student is a numerical attribute assigned to each student depending on his/her performance in the examination. The most capable student will be given the 1^{st} rank and consecutive ranks denote a decrease in capability. The lower the rank, the more capable the student. Ranks have been traditionally allotted based on marks scored in the examination.

The capability level referred to above is the 'raw' capability level that is required to be estimated to good accuracy using an exam. This paper proposes the maximum-likelihood estimation method, and we will later refer to this estimate of the capability as 'ML-capability' level. Now we describe the parameters that will be used to compare the reliability of our method with traditional marks based ranking.

- **Gate crashers**: These are the students who are actually of lower capability but get ranked higher into a group of more capable students due to various factors (mentioned in Section 1). The groups we refer to will be various percentages of the most capable student population. This metric is analogous to finding the number of students who are undeserving cross an eligibility criteria, which is a frequently used metric in many examinations.
- **Maximum rank jump/drop**: This is the difference between the actual rank of a student and the rank allocated after evaluation. If a less-capable students gets a higher rank, then we call it as a rank jump. If a student with higher actual capability gets allocated a lower rank when compared to that student's actual rank, we call it as a rank drop. Both, the rank jump and the rank drop, can occur when ranks are allocated on basis of ML-capabilities or marks. This parameter brings to our notice a long term effect of ranking errors. A student who gets eligible because of a rank jump is likely to be disappointed when he fails to perform as well as his classmates. Rank jump also effects the learning and classroom experience of a complete batch of students as instructor has to consider this less capable student group. Similarly, rank drop causes capable students to miss out on opportunities to study in reputed institutions and may stagnate their academic progress.

In the next section, we look at the relation between the parameters defined here and how they are used to decide a student's response to a given question.

2.1. The relation between confidence level, capability level and difficulty level

In this section we model the confidence level of students while answering questions of various difficulty levels. As described in the previous section, confidence levels depends on both capability and difficulty levels. This relation models student behavior in answering examinations and how it is affected by the difficulty level of questions. This relation can only be estimated from data of previous examinations. The most accurate way to find the form of this relation would be by using data from past CAT based examinations. As both capabilities and difficulty levels would be known quantities, we could easily find the form these curves would ideally take. However, currently we do not have access to such data. So, an approximate model (shown in Figure 1) is described below and we use this in our study. The confidence level (probability of getting answer correct) depends on difficulty level d and the capability of a student c as follows.

$$\text{Confidence level} := \frac{e^{a(c-d)}}{1+e^{a(c-d)}} \qquad (1)$$

The parameter a (in equation (1)) decides the slope of the curves and depicts how well a question can differentiate between students' capabilities. Ideally, the slope should be infinite, that is, a vertical line which divides the students into two groups based on their capabilities. However, due to the factors mentioned earlier, these curves have finite slopes at all points. It is assigned a constant value of 50 in this paper.

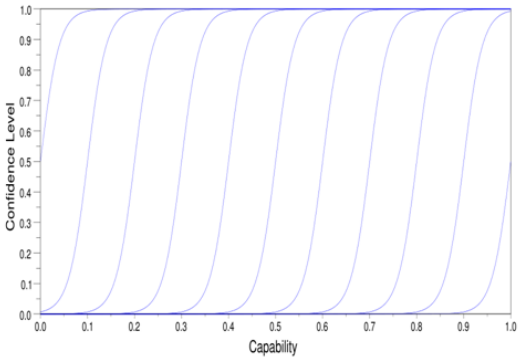

Figure 1. Confidence level vs easiness level for ten capability levels: 0.1, 0.2,...,1. The left-most curve is for difficulty level of 0.1, and the right-most curve is for difficulty level of 1

On inspection of Figure 1, we see that the guessing phenomenon has been ignored (as stated in Section 1). If guessing were to be considered, then we would be seeing an upward offset in the curves, providing each student with a fixed probability a getting a question correct, irrespective of the question's difficulty and the student's capability. We will now describe the proposed algorithm for estimation of ML-capabilities in the next section.

3. An iterative algorithm for assigning difficulty/capability levels

We motivate briefly the significance of the proposed algorithm and the intuitive reasons. First consider the case when all questions have equal difficulty level (and correspondingly equal weightages). In such a case, the total marks obtained by a student indicate the capability level. The drawback of this is that more weightage should ideally be assigned to a question which has been answered correctly by a smaller number of students. This calls for first setting the question-wise difficulty level, which can be achieved by setting the difficulty-level high if less number of students get a question right. This however amounts to the 'dual' drawback of giving equal weightage (i.e. assuming equal capability levels) for all students.

Given the above drawbacks of one-time assignment of difficulty/capability levels, we propose an iterative algorithm that alternates between capability level assignment and difficulty level assignment: each time using a maximum likelihood based method. Such 'alternating' iteration schemes/heuristics have been studied widely in the literature and these heuristics have only recently been supported with performance guarantees: see [10]. Work reported in the literature has been primarily in low rank approximations and not much in the problems concerned in this paper.

3.1. The marks matrix

As described above, this paper aims to assign difficulty levels (and hence weightages) to questions based on aggregate students' question-wise performance. Hence, we start with a so-called marks matrix: a matrix whose columns are indexed by questions and the rows indexed by the students. Each entry represents the student's performance in that question: this is typically either zero or one, or a suitable negative value in case of negative marking. In this paper, we do not consider negative marking.

3.2. Maximum likelihood based iterative method

A typical marks matrix is shown below. Rows correspond to students and columns correspond to questions.

$$\begin{array}{cc} & \text{Easiness} \\ & \begin{array}{cccc} e_1 & e_2 & \dots & e_m \end{array} \\ \text{Capability} & \begin{bmatrix} 1 & 1 & \dots & 0 \\ 1 & 0 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \dots & 0 \end{bmatrix} \begin{array}{l} c_1 \\ c_2 \\ \vdots \\ c_n \end{array} \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix} \\ & \begin{bmatrix} q_1 & q_2 & \dots & q_m \end{bmatrix} \end{array}$$

n : Number of students who have taken the exam.
m : Number of questions in the question paper.
c_i : Capability level of the i^{th} student.
d_j : Difficulty level of the j^{th} question.
m_i : Total marks scored by the i^{th} student.
q_j : Number of students who answered the j^{th} question correctly, in other words, total marks scored by the j^{th} question.
C : Student capability vector = $[c_1...c_n]$
D : Question difficulty vector = $[d_1...d_m]$
m_c: Marks that could be obtained by a student of capability c for given D.
q_d : Number of students who might answer correctly given question's difficulty level d and C.
The inputs to the algorithm are the total marks vector $M=[m_1...m_n]^T$ and question total marks $Q=[q_1...q_m]^T$ whose entries are column-wise sums and row-wise sums of marks matrix respectively. The algorithm outputs $[c_1...c_n]$ and $[d_1...d_m]$. The output $[c_1...c_n]$ is the ML-capability level, i.e. the capability level estimated by the maximum likelihood estimation procedure described below. A pseudo-code of the algorithm is presented. At Steps 5 and 9, maximum likelihood aspect of the algorithm comes into play.

We give a brief explanation about the pseudo-code here. We are carrying out successive minimization over two variables which are interdependent i.e., difficulty levels d and capability levels c. We start with an initial value for D in step 1. Over steps 4 and 5, we use these arbitrary difficulty levels for the questions and find the value of capability c_i of each student such that his actual marks m_i are as close as possible to the most likely marks m_c that he could obtain.

Algorithm 1 Pseudo-code of the iterative algorithm

1: Initialize D
2: **for** Fixed number of iterations **do**
3: **while** Error norms of estimated levels in previous iterations $\geqslant$ tolerance value **do**
4: **for** Each student i **do**
5: Given D find $c \in [0,1] \mid (m_c - m_i)^2$ is minimum.
6: $c_i \leftarrow c$
7: **end for**
8: **for** Each question j **do**
9: Given C find $d \in [0,1] \mid (q_d - q_j)^2$ is minimum.
10: $d_j \leftarrow d$
11: **end for**
12: $D \leftarrow \frac{0.5 \times D}{\text{mean (D)}}$ ▷ Normalization of difficulty vector
13: Go to Step 4
14: **end while**
15: **end for**

We store all these estimates of capabilities in the vector C in step 6. This is the first minimization. In the second minimization, done in steps 8 and 9, where we perform a similar operation to find out the difficulty level d_j of each question such that the number of students who answered it correctly q_j is as close as possible to the number of students who are most likely to answer it correctly q_d. We store the estimates obtained for question difficulty levels in D, and normalize its mean to 0.5 in step 12. Normalization is done to make the algorithm insensitive to initial estimate that may be assigned to D. We then use this estimated difficulty vector and run the algorithm again from step 4. The break condition for this iterative procedure is that error norms between estimates over successive iterations becomes lower than a fixed tolerance level.

4. Simulation and numerical experiments

In this section we perform simulation studies with the objective of comparing the efficiency of ranking of students based on marks with ranking based on ML-capabilities level. The simulation study demonstrates that our proposed algorithm provides more accurate ranking than the traditional marks based ranking. Our simulation setup is as follows: We start with N students who solve a question paper with M questions. In our first step, we assign capability levels to the students and difficulty levels to the questions. Then, for each student we generate a vector indicating the marks obtained in each question. Note that the i^{th} student gets the j^{th} question correct with probability p_{ij} , indicating his confidence level which is known as c_i and d_j are known. Once the marks are obtained, they are given as input to our proposed algorithm. It is important to note that algorithm only knows the marks matrix and has no knowledge of the assigned (actual) capability and difficulty levels. The algorithm has been shown to converge to the actual capability and difficulty levels with acceptable accuracy in [9]. Once the break condition (described in section 3.2) occurs, we get the estimated capabilities (ML-capabilities) and difficulty levels. To gauge the performance of our algorithm, we compare the student ranking based on ML-capabilities with the student ranking based on actual values used to generate the marks matrix. To eliminate the randomness that occurs while generating the marks matrix, we perform 100 simulations and look at the averaged over results. The simulations are carried out in Scilab: [8]. We now explain our procedure to obtain a marks matrix by using assigned capabilities and difficulty levels.

4.1. Student Response Simulation

In this section we explain in detail the method we employ to generate the marks matrices used during our simulations. As it is not feasible to conduct 100 tests with real students, we need a method to generate marks matrices which are a close approximation to an actual marks matrix. To this end, we propose the "simAttempt(·)" function to simulate student attempts in an examination to answer a question.

4.1.1. The simAttempt(·) function. The confidence level of a student for a question decides if the student is able to answer the question correctly.

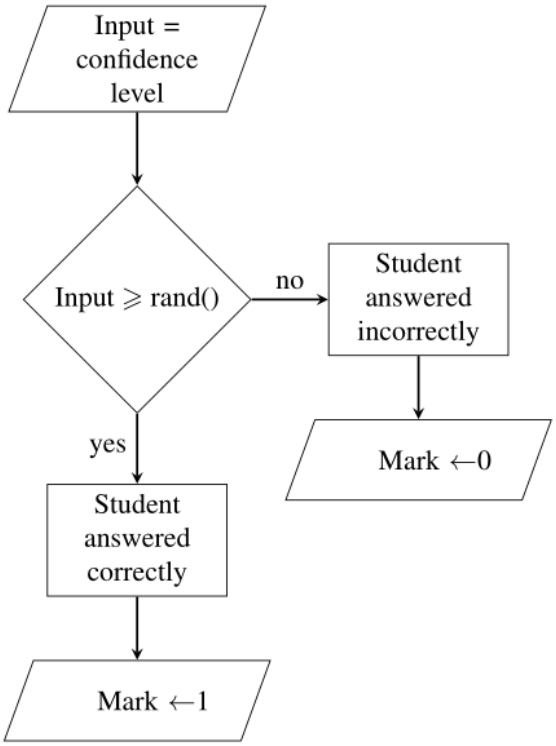

Figure 2. simAttempt(·) function: flowchart

As was defined in Section 2, confidence level is the probability of getting a question of given difficulty level correct by a student of a given capability. Being defined as a probability, it implicitly means that it is a ratio of expected outcomes to the number of all possible outcomes. For example, when we say that the probability of getting a heads on a coin flip is 0.5, it means that if the coin is flipped n number of times, then we get heads for n/2 times. Similarly, when a student has a confidence level of 0.29 for answering a particular question, it means that, if the student attempts 100 questions of the same difficulty level, the student secures a score of around 29 out of 100.

To implement a function which generates marks based on the confidence level such that it satisfies the intuitive explanation given above, we use the cumulative distribution function of a uniformly distributed random variable. Flowchart of the code is given in Figure 2. The rand() function is configured to generate random numbers with uniform distribution between 0 and 1.

4.2. Rationale behind the simAttempt(·) function

Figure 3 shows the cumulative distribution function
(CDF) of a uniformly distributed random variable. The CDF is given by the equation

$$F_X(x) = \mathrm{P}(X \leqslant x). \qquad (2)$$

$F_X(x)$ is the probability of the random variable X being at most the value x. For uniform distribution, as can be seen, it is a linear increasing function taking the values between 0 and 1 as it denotes probability. Therefore, the probability that the random variable has a value of 0.2 is less than the probability of the random variable having a value of 0.6. By considering confidence level as a random variable, if it has a higher value then the probability of it being greater than random number generated by the rand() function is higher. Hence a higher confidence level gives a higher probability of a student getting that question correct.

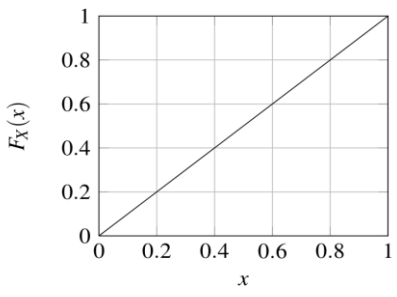

Figure 3. Cumulative distribution function for a uniformly distributed random variable with values between 0 and 1

The simAttempt(·) function was tested for various number of questions and the outcomes have been tabulated in Table 1. Q denotes the number of questions, column E shows the marks obtained for one execution and the column AVG shows the marks averaged over 10 executions.

Table 1. Marks obtained using simAttempt(·) for various confidence levels, single execution (column E) and averaged over 10 executions (column AVG)

Confidence level	Marks scored					
	Q = 10		Q = 50		Q = 100	
	E	AVG	E	AVG	E	AVG
0.17	0	1	10	11	21	17
0.45	4	5	29	22	49	42
0.91	10	8	50	46	88	91

It may be noted that simAttempt(·) does not always give the exact marks expected from the input confidence level. However, on an average, the marks scored is indicative of the confidence level. Hence, as the number of questions increases we have a better opportunity at capturing confidence levels. However, in any examination, the number of questions is low. simAttempt(·) is therefore successful in capturing the nature of students committing mistakes in questions they are capable of answering. Thus, simAttempt(·) is able to simulate the student response in the exact way we wanted it to. In the next section, we describe the simulations we performed using the marks matrices generated by using the method that was explained in this section.

4.3. Simulation Results

In this section we discuss the results we obtained on performing various simulation studies. We first consider the gate crashers (see section 2) among top 10% of the students. The simulation setup described in section 4 is conducted for student populations of 1000, 2000, 3000, 4000 and 5000 answering a test with 60 questions. The number of gate crashers averaged over different realizations of the marks matrices has been shown in Table 2. It can be seen that ML-capability based ranking performs better at reducing the number of gate crashers than what was possible by mark based ranking. To get a better perspective about this result in large scale examinations, we extrapolate and find the number of gate crashers if 125,000 students appear for an examination. If students are ranked based on raw-marks, we would get around 1120 gate crashers. This is unacceptably high when one considers the situation where the number of seats available in highly reputed institutions is much lower than those appearing for the exam. However, if one ranks based on ML-capabilities, we get around 216 gate crashers

which is a very good improvement over 1120 gate crashers.

Table 2. Comparison of gate crashers in the top 10% using ML-capability level and student marks

Number of students	ML-capability based	Marks based
1000	6	7
2000	17	15
3000	16	23
4000	28	33
5000	29	37

But, both the schemes have allowed gate crashing. We focus on the gate crashing students who were allowed by these schemes. Among the gate crashing students shown in the Table 2, we look how many gate crashed in both schemes and how many gate crashed in each of the individual schemes. This gives an idea about the how much gate crashing the schemes may be allowing. Consider Table 3. Column A contains the number of gate crashers allowed by both schemes. Column B shows the number of students gate crashing due to the use of ML-capabilities alone, and Column C gives us the number of gate crashing students due to the use of marks based ranking alone. Comparing Column B and Column C we can say that marks based ranking has more allowance for gate crashing than ML-capabilities based ranking. In other words, if some amount of gate crashing was inevitable, ML-capabilities based ranking would fare better than marks based ranking.

Table 3. Comparison of allowance of gate crashing phenomenon by both ranking schemes

Number of students	Column A	Column B	Column C
1000	2	3	5
2000	9	10	19
3000	6	12	14
4000	11	16	22
5000	12	16	25

Now, we look at the number of gate crashers among various top percentages. This will give us an idea of how varying the eligibility criterion effects the number of undeserving students who qualify. Figure 4 shows this variation for a student population of 5000 students. The solid line shows the variation when ranking is based on marks and the dashed line shows the number of gate crashers for ranking based on ML-capabilities.

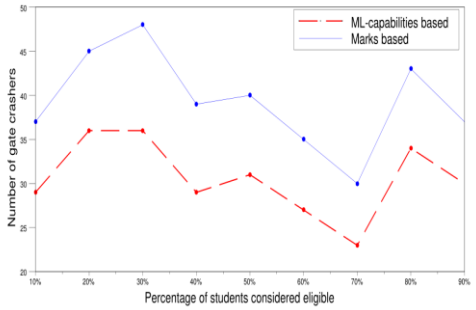

Figure 4. Comparison of gate crashing among various top percentages among a student population of 5000

It can be seen that an appreciable margin exists between the two. Hence ML-capabilities based ranking does better over all the percentages that may be chosen as cutoff in choosing eligible students. Next, we will perform rank jump and rank drop analysis on the results of our simulation. Suppose a ranking scheme is able to minimize the number of gate crashers to a very low number. In such a case, if the rank jump and rank drop are still high, then the scheme could be considered a failure. Recall that, ranking determines if a student is able to get his choice of institution (see section 1).

We now look at the errors in ranking of students by both the schemes. This comparison of ineligible jumping into higher ranks and eligible students getting dropped into lower ranks is documented in Tables 4 and 5.

Table 4. The maximum rank-jump into the top 10th percentile group

Number of students	ML-capability based	Marks based
1000	27	36
2000	74	97
3000	133	146
4000	122	159
5000	200	236

Table 5. The maximum rank-drop from the top 10th percentile group

Number of students	ML-capability based	Marks based
1000	39	45
2000	84	108
3000	153	164
4000	165	186
5000	231	300

Therefore ML-capabilities based ranking scheme performs better in avoiding these errors than the

traditional scheme in both the cases. In the next section we study the algorithm's behavior when our assumption about the form of confidence level curves is challenged.

4.4. Robustness of algorithm

In this section, we look at effect of varying the dependence of confidence level on capabilities and difficulty levels and study the reliability of the ML-capabilities based ranking scheme. The approximation of student answering behavior depicted in Figure 1 is a key assumption in our study. As stated in section 2.1, the relation was formulated by our experience with students and exam evaluation. The actual relation can only be estimated by analyzing data from previous examinations or from examinations conducted in a controlled environment. Thus, it is possible that the actual relation varies from ours by a lesser or larger margin. We look at the values of the performance metrics by perturbing the curves we used earlier. In Figure 5, the solid line represents the original relation and the dashed line represents a variation.

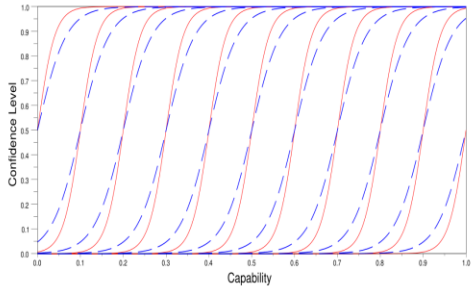

Figure 5. Perturbed confidence level curves; solid line denotes the original curves, dashed line shows the perturbed curves

The simulations are run for 1000 students, 60 questions and the results given in Table 6. The results presented are an average over 100 tests simulated by using the same student population and question set, similar to experiments in the previous section.

Table 6. Sensitivity of algorithm to changes in student response characteristics

Parameter '*a*' →	40	45	**50**	55	60
Gate Crashing Students	7	7	**6**	7	5
Average Rank Jump	13	10	**9**	12	13
Average rank Drop	13	13	**12**	13	13

From Table 6, we can see that the performance metrics are about the same value as the original one (when 'a'= 50) for small changes in the parameter 'a'. Thus, our algorithm is robust enough to changes in parameter 'a' of equation (1).

6. Concluding remarks and future work

In this paper we proposed a way to analyze off-line examinations for the problem of aposteriori assignment of question difficulty levels and student capability levels. The analysis involved a systematic way to simulate student response and a maximum likelihood based method to estimate student capabilities and question difficulties based on the marks obtained. The maximum likelihood based capability estimation (ML-capability levels) is a better value to use for ranking, instead of the traditional raw marks based method. We have also clearly shown that, ineligible students can gate crash into the group of more capable students, even if guessing is made impossible in examinations. Therefore, in both cases i.e., with or without guessing, ranking based on marks is erroneous and therefore not reliable.

We proposed a method to assign the difficulty levels of the questions and the capability levels of the students from the marks scoredby the students. It can be inferred from the data in various tables is section 4.3, that ranking based on ML-capability levels is much more accurate than traditional marks based ranking. The key feature of CAT, namely, a question is termed harder (and hence carries greater weightage) when less students get the question right, is incorporated in the proposed off-line method, in spite of the aposteriori aspect of the algorithm. We also studied our algorithm for small changes in the student confidence level behavior and showed that it is robust to such changes. We performed numerical experiments in Scilab for a situation that is typical in the GATE exam (see [7]): a very small fraction of students are selected from those appearing for the exam, and secondly, elimination of guessing is possible due to exact numerical answer format instead of multiple-choice format. However, our approach has the following key limitations:

- The relation between confidence, capability and difficulty levels is not an exact one.
- We have not incorporated the possibility of guessing which is one of the key factors that introduces errors in ranking.
- Our ranking scheme still allows some gate crashers. This is a major drawback and has to be countered.

Possible directions of further investigation include convergence analysis of the algorithm. Numerically efficient algorithms need to be employed for quicker implementation of the proposed algorithms. An

eventual objective of this work is to validate the algorithms on large-scale real exam data and, after perhaps some fine-tuning, use ML-capability based methods to rank students instead of the traditional total raw-marks based methods.

7. Acknowledgements

This work has been supported in part by SERB, DST and BRNS, India.

8. References

[1] E. Lesage, M. Valcke, and E. Sabbe, "Scoring methods for multiple choice assessment in higher education–Is it still a matter of number right scoring or negative marking?," *Studies in Educational Evaluation*, vol. 39, no. 3, pp. 188–193, 2013.

[2] R. L. Karandikar, "Multiple-choice tests, negative marks and an alternative," *Resonance*, vol. 11, no. 3, pp. 86–93, 2006.

[3] R. L. Karandikar, "On multiple choice tests and negative marking," *Current Science*, vol. 99, no. 8, pp. 1042–1045, 2010.

[4] F. B. Baker, *The Basics of Item Response Theory*. Education Resources Information Center Publications, 2nd ed., 2001.

[5] D. C. Briggs and M. Wilson, "Generalizability in item response modeling," *Journal of Educational Measurement*, vol. 44, no. 2, pp. 131–155, 2007.

[6] T. H. Wang, K. H. Wang, W. L. Wang, S. C. Huang, and S. Y. Chen, "Web-based assessment and test analyses (WATA) system: development and evaluation," *Journal of Computer Assisted Learning*, vol. 20, no. 1, pp. 59–71, 2004.

[7] GATE, *The Graduate Aptitute Test in Engineering*. IITs and IISc, India, 2015.

[8] Scilab Enterprises, *Scilab: Free and Open Source software (FOSS) for numerical computation*. France, 2012.

[9] V.G.Padmaji, S.Cheruku, P.Chaporkar, and M.N.Belur, "A maximum likelihood based computation of student-capability and question-easiness levels," in *Proceedings of the Ireland International Conference on Education*, Dublin, Ireland, 2015.

[10] P. Jain, P. Netrapalli, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, STOC '13, pp. 665–674, New York, USA, 2013.

