DOCUMENTO DE DISCUSIÓN

DD/22/15

Implementation of a Computerized Assessment System by using Backpropagation Neural Networks with R and Shiny

Juan Manuel Gutiérrez Cárdenas (Universidad del Pacífico) jm.gutierrez@up.edu.pe

Fernando Casafranca Aguilar (Universidad del Pacífico) casafranca_af@up.edu.pe

Diciembre, 2015

Implementation of a Computerized Assessment System by using Backpropagation Neural Networks with R and Shiny

Juan M. Gutiérrez Cárdenas and Fernando Casafranca Aguilar
Departamento Académico de Ingeniería, Universidad del Pacífico
Lima, Perú
(JM.GutierrezC, Casafranca_af)@up.edu.pe

Abstract

The discouragement, that early undergraduate students suffer when they are faced to topics that they struggle to master, could increase owing by the use of inadequate evaluation materials. It is generally found that in the classroom there are students that manage to cope with the material of the courses in a quick manner, while others present difficulties while learning the material. This situation is easily spotted in the examination results, a group of students could get good marks encouraging them to tackle the course optimistically while others would get the wrong perception that the topics are difficulty, and in some cases, forcing them to leave the course or in other cases to change careers. We believe that by the use of machine learning techniques, and in our case the utilization of neural networks, it would be feasible to make an evaluation environment that could adjust to the needs of each student. The latter means that the system could auto tune the difficulty of the given questions to the students, allowing a more dynamic evaluation system which at the end would decrease the feeling of dissatisfaction and drop off the courses.

keywords: Computerized Assesment Systems, Computer Science Education, Neural Networks

1 Introduction

Computer Adaptive Testing or CAT is the use of Computer Science techniques and algorithms that could be helpful for the evaluation of students. The effort of developing these software tools has a wide range of applicability from e-learning to distance learning and even for the use inside the classroom. CATs are used for two purposes mostly: 1) To deal with a massive number of students and 2) To obtain automatic feedback and act in a personalized fashion for each student. We believe that by using a CAT aggregated with the prediction possibilities derived from the use of a neural network, that it would be possible to tune the difficulty of an examination applied to a student. These automatic tuning will allow that the student would fell unstressed of the pressure of coping with topics that he is still not able to handle; and at the same time it could be a good indicator of those sections where the student would need reinforcement by the lecturer.

This paper is organized in the following way: in Section 2 we make a description about Automatic Assessment Systems and how they have developed in different institutions, in Section 3we will describe Neural networks and specifically the Backpropagation Algorithm that we use for our proposal, in Section 4 a description of our proposal along with the results obtained for applying it in our institution are showed.

2 Automatic Assessment Systems

Amruth [1] made an automatic evaluation to be used in the marking of programming tasks by using templates. In his schema the templates, which has similarities to a lexical analysis used in Formal Languages, allow to increase the difficulty, variety and non-redundancy of the generated questions. The system can also mark the questions submitted by using a tree traversal approach. Different techniques such as: Neural Networks, Machine Learning or Agents can be used to tune the difficulty of the problems given to the students. Nevertheless the goal is not to make these tools based only on AI methods, but to also assess the impact in the learning outcomes of the students [2].

Some CAA tools are based in existing software, for example in the Algorithms and Data Structures online tool made by Grcia-Mateos and Fernndez-Alemn [3] an extension was made to the Mooshak learning system [4]. The Mooshak framework was an environment created for the automatic testing in programming situations by using a web server architecture; framework that was extended and applied to students evaluations as we mentioned beforehand. Among the pedagogical recommendations that Garca and Fernndez [3] stated in their article, we choose a pair of them that seems rather convenient: motivation and active learning. Motivation related to strive to minimize the number of students that drop a subject; and active learning so that the student is aware of its own advance in the subject.

The application of online assessment techniques could not have immediate responses when applied to students group in Computer Science as stated by Wolf and Manson [5]. In their research work they had to tune the right amount of online sessions that a set of pupils should have, before showing promising results. An increase of confidence, less plagiarism cases and strong correlation between topics learned and assessed are the results of a well managed online evaluation tool.

A complete survey about the different automatic evaluation tools in the period from 2006 to 2010 can be found in the work of Ihantola et al. [6]. The authors establish a division between two trends that are commonly found in this type of systems: one is the marking of programming questions for programming contests; the other is the use for lecturing topics related to programming. Furthermore some recommendations about how to make the development of these systems more stable are stated. Among these recommendations, for example, is related to the inclusion of plug-ins, which could help to include these evaluation tools into existing online classrooms environment or LMS, Learning Management Systems, such as Moodle. Additionally it remarks the need to make these computer programs open-source oriented; in a way that the improvements over the existing software could advance in a quicker fashion. Moreover, in the research made by Pears et al. [7] we can find reasons which explain why the spread of these tutoring and evaluation systems is rather limited. For example, the local focus of these software tools aimed to improve the situation within a particular academic institution; or their development, as part of funding or graduate research projects. With these limitations there is impossibility that these works would continue their development after their presentation.

Different types of subjects could be taught by using an online assessment tool, and Computer Science is not the exception. Topics from Algorithms and Data Structures [3] algorithms and preparation for programming contests [8], programming tasks [9,10,11] math and other topics [12,13,14] are available for helping the educational process. The tuning difficulty of the questions oriented to the students of these type of software are made by using a fixed schema, or using internal engines based mostly in Artificial Intelligence (AI) techniques such as: Neural Networks, Machine Learning or Agents [2, 15].

A system which employed a combination of neural networks and an expert system was also proposed [16]. In this software tool a backpropagation neural network is trained with the results of previous examinations stored in log files; the neural network will allow to predict if the student will have difficulties or not in the topics that will be assessed according to an established threshold. When the difficulty level was predicted, then an expert system brought the question to the student for its examination.

The students that a lecturer faces in a Computer Science group are dissimilar in their intellectual productivity and capabilities. Ranging from those students that can understand, abstract, analyze and implement new code based on the taught topics easily; and students that only achieve little tasks within the scope of the objectives of the course. In Lister and Leaney [17] the authors proposed a schema based on the Bloom taxonomy so tests are diversified and each group of students achieves their goals. Within that schema there is an increase in the number of students that pass the course, but no diminishing in the quality of the topics given to them. Additionally that is one of our main goals that we propose in our research: Diversify the different levels of examinations aimed to different types of students in a classroom.

The central point that we aim to solve in our research is how to establish a method for increasing, gradually, the difficulty of a set of problems given to a student. There have been some successful attempts e.g. Khan Academy [18] based on a regression technique [19]. The GRE examinations [20] also use an adaptiveness feature, even though is not clearly documented. About the submissions schema that our proposed system could handle they could be oriented to multiple choice and essay questions.

This special type of software, that modifies the difficulty of questions based on the previous answer from an examinee are called Computerized Adaptive Testing (CAT) (Weiss, 2004). This technique is so popular that it has also its own Association called IACAT (2009) or International Association for Computerized Adaptive Testing.

3 Neural Networks

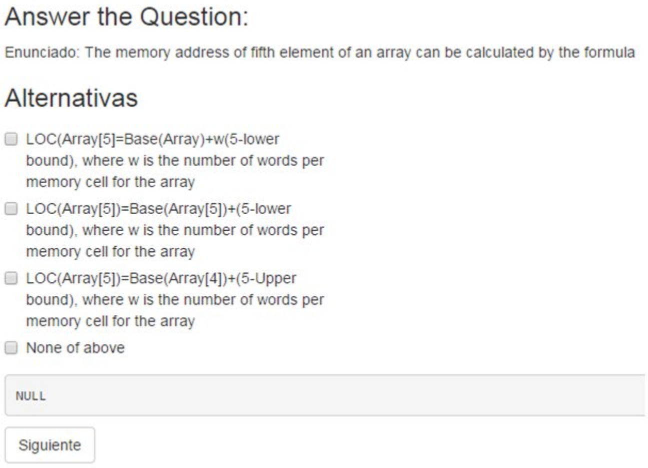
We will describe in this section the part concerning to Artificial Neural Networks and the Backpropation algorithm which will serve as a background for our proposal. Most of the material in this section is well described in the literature available, but we have based most of this part by considering the approach that Tom Mitchell states in its book of Machine Learning [22].

Artificial neural networks have their basis on the biological model, where we have a set of neurons that are connected one thru another to build a neural net. In the artificial model the neurons are also connected as a set of nodes that are influenced by a set of weights; each neuron has an activation function that allows the information to pass from one node to another. The simplest model was a single layer neural network, ie. having only one input, a neuron or set of neurons and an output; this model was known as the Perceptron model. The inability of this model to handle non-linear separable problems arise the need of the use of connection of neurons in a multilayer fashion and this model is known as the multilayer Perceptron model.

When one neuron receives an input, and according to its activation function, propagates the information to other neurons is named as the Forward Propagation Algorithm. Drawbacks of this model, as the impossibility of update the weights that influence each network gave birth to what is knows as the Backpropagation algorithm. In this latter model after performing a Forward Propagation algorithm the weights are updated by making a reverse traversal thru the nodes of the Neural Network, by diminishing the error that is obtained by comparing the actual output to the needed output in the learning process phase. In our proposal we have used a model of Multilayer Perceptron and by using the Backpropagation algorithm for training and make predictions with our neural network model.

4 Methodology

We have implemented a web application by using the R language and the Shiny web development framework. A representation of the interface can be seen on Figure 1:



Answer the Question:

Enunciado: The memory address of fifth element of an array can be calculated by the formula

Alternativas

□ LOC(Array[5]=Base(Array)+w(5-lower bound), where w is the number of words per memory cell for the array

□ LOC(Array[5])=Base(Array[5])+(5-lower bound), where w is the number of words per memory cell for the array

□ LOC(Array[5])=Base(Array[4])+(5-Upper bound), where w is the number of words per memory cell for the array

□ None of above

NULL

Figure 1: Draft of the interface presented to a student for an exam submission. A question is visualized and the student can mark one or more alternatives, the later one in case that the question will have multiple answers. The question is set up according to the difficulty given by the neural network; and when the user press the button Next he is given another question with the tuned level of difficulty according to how well does he made it on the former question.

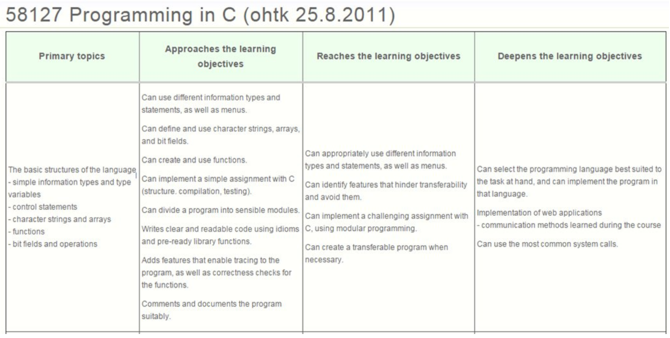
Our scheme is simple: The student is presented with a chosen question according to the level of difficulty given by the neural network. We have chosen five different levels of difficulties considering three main levels and two mid-difficulty levels. The main levels were considered taking into account the recommendations for creating Learning Objectives Matrices given by the Department of Computer Science of the University of Helsinki [23]:

Level 1: Approaches the Learning Objectives

Level 2: Reaches the Learning Objectives

Level 3: Deepens the Learning Objectives

It is mentioned that a student who reaches the first level should be considered to pass the course, while a student that reaches the second level should be acquainted with maximum mark obtained for a course; the last level comprises advanced topics that are not commendatory for the student to achieve [23]. In figure 2 we have depicted the aforementioned levels for the case of a C programming course:



58127 Programming in C (ohtk 25.8.2011)

Primary topics	Approaches the learning objectives	Reaches the learning objectives	Deepens the learning objectives
The basic structures of the language - simple information types and type variables - control statements - character strings and arrays - functions - bit fields and operations	Can use different information types and statements, as well as menus. Can define and use character strings, arrays, and bit fields. Can create and use functions. Can implement a simple assignment with C (structure. compilation, testing). Can divide a program into sensible modules. Writes clear and readable code using idioms and pre-ready library functions. Adds features that enable tracing to the program, as well as correctness checks for the functions. Comments and documents the program suitably.	Can appropriately use different information types and statements, as well as menus. Can identify features that hinder transferability and avoid them. Can implement a challenging assignment with C, using modular programming. Can create a transferable program when necessary.	Can select the programming language best suited to the task at hand, and can implement the program in that language. Implementation of web applications - communication methods learned during the course Can use the most common system calls.

Figure 2: Learning Approaches Matrix [23] for a Programming in C course. When the student approaches the learning objectives he gains the minimum passing grade for the course, and when he reaches the Learning Objectives his grade is the maximum that can be obtained for that given course. The advanced or mastering of the topics is obtained when the student Deepens the Learning Objectives. The verbs follow the Bloom Taxonomy.

In our proposal, we will have the same levels mentioned in [23], but with two mid-difficulty section between levels:

Level 1: Approaches the Learning Objectives, the student is able to master very basic topics.

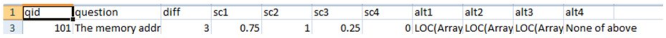
Mid-Level 1-Level 2: Questions in this level are between basic concepts and the master of some topics without reaching an intermediate level.

Level 3: Reaches the Learning Objectives; this is what we consider the intermediate level where the student is comfortable with the lectured topics, and can devise solutions using the learned techniques.

Mid-Level 2-Level 4: The questions put on this level approaches to the level where a student could show enough expertise than the average of the topics covered during the course, but reaching the point where he is able to propose new solutions by using the combination of different topics given during the course.

Level 5: Deepens the Learning Objectives, the student is able to deduct solutions for problems where only an insight or a very brief explanation was given during the course.

The questions are stored in a file grouped according to the number of the question ie. create a subset of questions, for example 10, that cover question 1, another subset of 10 questions for question 2; and so on. In this way, we secure to minimize the number of repeated questions for a group of students with the same difficulty level. Each question is stored in a CSV format for easy access as we can observe in Figure 3.

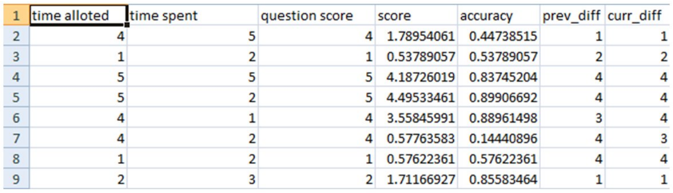


	qid	question	diff	sc1	sc2	sc3	sc4	alt1	alt2	alt3	alt4
1	qid	question	diff	sc1	sc2	sc3	sc4	alt1	alt2	alt3	alt4
3	101	The memory addr	3	0.75	1	0.25	0	LOC(Array	LOC(Array	LOC(Array	None of above

Figure 3: Each question has its corresponding id where the first digit represents the number of the question. A difficulty level is setup by the lecturer as long with the individual scores per each alternative chosen, and the written alternatives that will be displayed to the student at the moment of his examination.

We considered that each question should be answered in a determined allotted time, which corresponds to the score of that individual question. For example, if a question has a score of 5 points, that means that it should be answered in five minutes approximately; we believe that by setting the points related to the time spent on answering it will make it easier for the lecturer to assign the scores per each question.

We have implemented a neural network model that by using randomly generated data will allow the automatic difficulty tuning per each question; for this purposes we have considered that if a student answers a question within the time given and if the question has been answered correctly, with an accuracy greater than 50%, then he should be promoted to the next level of difficulty; otherwise he will be penalized by lowering the difficulty of the upcoming question. The accuracy factor is obtaining by dividing the score obtained by the student by the score that was given by the lecturer to a specific question. All the data has been generated randomly to simulate the behaviour of a student in answering a given question with a certain level of difficulty as we can observe in Figure 4.



1	time alloted	time spent	question score	score	accuracy	prev_diff	curr_diff
2	4	5	4	1.78954061	0.44738515	1	1
3	1	2	1	0.53789057	0.53789057	2	2
4	5	5	5	4.18726019	0.83745204	4	4
5	5	2	5	4.49533461	0.89906692	4	4
6	4	1	4	3.55845991	0.88961498	3	4
7	4	2	4	0.57763583	0.14440896	4	3
8	1	2	1	0.57622361	0.57622361	4	4
9	2	3	2	1.71166927	0.85583464	1	1

Figure 4: Random data generated that simulates a student response given a question with a certain level of difficulty. Time allotted is the time that is permitted for the student to solve the problem at hand, that value has a direct relationship with the score assigned to the question. Time spent is the time that the student took for answering the question. The accuracy level is calculated based on the ratio between the score obtained by the student and the maximum score assigned to that question by the lecturer. The last columns named previous difficulty and current difficulty, gives the previous difficulty level that the student was able to master, while the current difficulty level is the calculated one based on the data presented in the aforementioned described columns.

With the random data that gives us a glimpse of how the current difficulty could be generated in response to the type of question, time given, time spent, score and previous difficulty level achieved by the student is that we proceed to train a neural network by using the backpropagation algorithm. It is worthy to mention that even though the score of the question has a direct relationship with the time given to a question to be answered, that does not need to be followed like a fixed ruled, because the lecturer could decide that both variables are not directly related. Our trained neural network proposal has four inputs, two hidden layers of six neurons each and an output that gives us the current difficulty level or predicted difficulty for the next question to be answered by a student. We can observe our modeled neural network in Figure 5.

For testing our system we have developed a software tool by using R and its web environment tool known as Shiny. The course objective was an Introduction to Programming course; in this the lecturer presents the students the following topics:

a) Flow of Data

b) Basics of programming

c) Basic algorithm analysis

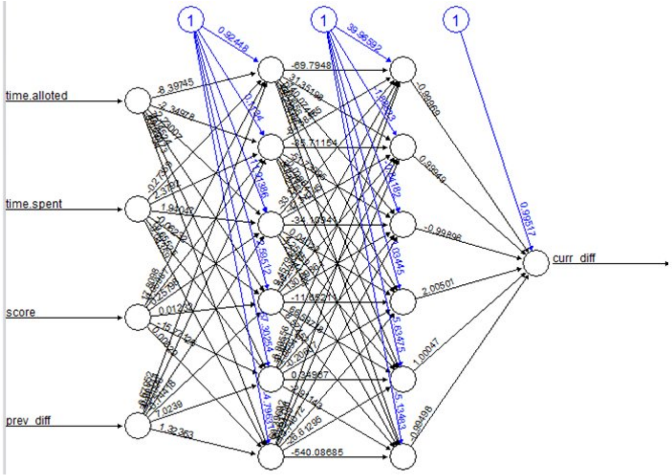


Figure 5: Neural network trained by the backpropagation algorithm. The inputs correspond to data related to the time given, time spent, score and previous difficulty of the later question given to a student. We manage to get an error of approximately 10E-4 in 76618 iterations. The accuracy attribute is not considered because it was used for generating the next difficulty level with the random data generated.

Table 1: General results obtained by applying our automatic assessment proposal to a group of 17 students in an introductory course of programming.

Number of,Students that made the examination	17
Number of,students who passed the examination	14
Number of,students who failed the examination	3
Students whom,improve their marks	6
Students,remaining in a general average level	8
Students whom,drop their marks	3
General,classroom average score over 4 points	2.18/4
General,classroom average difficulty obtained	3/5

Table 2: Number of students that approached, reached and deepened the learning objectives; as we can observe there was a decrease number of students who failed and an increase in those who approached and deepened the learning objectives. In general terms the automatic assessment system showed an increase in the learning objectives achieved by considering some intermediate and advanced topics put into the course syllabi.

	Traditional Assessment	Automatic Assessment
Approaches the learning objectives	6	11
Reaches the learning objectives	5	2
Deepens the,learning objectives	0	2
Failed	6	3

d) Searching and Sorting

e) Object Oriented Programming principles

The chose of Python as the tool where the students implement the above mentioned theoretical topics was because its easiness of use and light syntax over other programming languages. For our testing purposes we made a set of questions oriented to three main items:

a) Basics of programming: flow of control.

b) Algorithm analysis.

c) Object Oriented programming principles.

d) Implementation in Python.

The questions were put in a csv file in a multi choice format, and the student was prompted to give an answer or multiple answers depending of the question at hand. The results we obtained were the following:

In table 1 we show the relevant results of the applied automatic assessment system. This data was compared with the general marks obtained by the students while using a traditional assessment method, by this we refer to put to all the classroom to a same difficulty level, upgrading or lowering the difficulty in general in a subjective way according to how the students performed during the course. It was interesting to notice that from the three students evaluated who dropped their marks, they obtained a good score by using the traditional method; we hypothesize that this could be a direct result of not considering the following experimentation in a serious way or maybe because the difficulty level where they got good marks was a regular one.

Considering the traditional scoring method and based on the results obtained from the students, the class was divided into four groups: Failing students, those who did not approach the learning objectives (6 students); regular students, those who pass the course with the minimum grade and approached the learning objectives (5 students); good students those who reached the learning objectives (6) and outstanding students (0) those who deepened the learning objectives. The concepts of how the leaning objectives are met were mentioned and exemplified in Figure 2. In table 2 we present the results of comparing the traditional assessment vs the automatic assessment implemented according to the learning objectives defined:

5 Conclusions and Future Work

We have implemented an automatic assessment system by using a backpropagation neural network, considering a set of signals: difficulty level achieved, score applied to the question and time considered for solving the question. The training data was generated by using synthetic or artificial data created for our own purposes. Our system has showed an increase in the learning objectives achieved by the students by comparing them to the course contents where it was applied. As a future work it would be recommended to obtain other signals from the students who perform examinations during a course, such as stress signals among others.

6 References

[1] Amruth N. Kumar. Generation of problems, answers, grade, and feedback—case study of a fully automated tutor. J. Educ. Resour. Comput. 5, 3, Article 3, September, 2005.

[2] Corbett, A. T., Koedinger, K. R., Anderson, J. R. Intelligent Tutoring Systems. In: Helander, M., Landauer, T. K., Prabhu, P. Handbook of Human Computer Interaction. 2nd ed. New York: Elsevier. 849-874, 1997.

[3] Grcia-Mateos G., Fernndez-Alemn, J. A course on algorithms and data structures using on-line judging. SIGCSE Bull. Vol. 41, No. 3, 45-49, 2009.

[4] Leal, J. P., Silva, F. M. A.: Mooshak: a Web-based, Multi-site, Programming Contest System. Software-Practice, and Expertise, 33, 6, 567-581. 2003.

[5] Woit, D and Mason D. Effectiveness of online assessment. In Proceedings of the 34th SIGCSE technical symposium on Computer science education (SIGCSE '03). ACM, New York, NY, USA, 137-141. 2003.

[6] Ihantola P., Ahoniemi T., Karavirta V., and Seppl O. Review of recent systems for automatic assessment of programming assignments. In Proceedings of the 10th Koli Calling International Conference on Computing Education Research. ACM, New York, NY, USA, 86-93. 2010.

[7] Pears A., Seidman S., Malmi L., Mannila L., Adams E., Bennedsen J., Devlin M., and Paterson J. A survey of literature on the teaching of introductory programming. SIGCSE Bull. 39, 4, 204-223. 2007.

[8] Jordi Petit, Omer Gimnez, and Salvador Roura. Jutge.org: an educational programming judge. In Proceedings of the 43rd ACM technical symposium on Computer Science Education(SIGCSE '12). ACM, New York, NY, USA, 445-450. 2012.

[9] Kumar, A. N. Generation of problems, answers, grade, and feedback-case study of a fully automated tutor. Journal on Educational Resources in Computing (JERIC). Vol. 5, No. 3, Article 3. 2005.

[10] Mike Joy, Nathan Griffiths, and Russell Boyatt. The BOSS online submission and assessment system. J. Educ. Resour. Comput. 5, 3, Article 2. September, 2005.

[11] Saikkonen R., Malmi L., and Korhonen A. Fully automatic assessment of programming exercises. SIGCSE Bull. Vol. 33, No. 3, 133-136. 2001.

[12] Rasila, A., Harjula, M., & Zenger, K., Automatic assessment of mathematics exercises: Experiences and future prospects. In Yanar, A., Saarela-Kivimki, K. (Eds.), ReflekTori 2007 Symposium of Engineering Education, 70-80. Helsinki University of Technology. 2007.

[13] Angelo Tartaglia and Elena Tresso. An Automatic Evaluation System for Technical Education at the University Level. IEEE Trans. Educ.. Vol 45 (No.3), p268. 2002.

[14] Isotani, S., Brando, L. O., An algorithm for automatic checking of exercises in a dynamic geometry system: iGeom. Computers and Education. Vol. 51 (No. 3), pp. 1283-1303. 2008

[15] Joseph E. Beck and Beverly Park Woolf and Carole R. Beal. 2000. ADVISOR: A machine learning architecture for intelligent tutor construction. Proceedings of the National Conference on Artificial Intelligence, 552-557.

[16] Abu Naser, S. S., Predicting learners performance using artificial neural networks in linear programming intelligent tutoring system. International Journal of Artificial Intelligence & Applications (IJAIA), 3(2), 65-73, 2012.

[17] Lister R. and Leaney J. , Introductory programming, criterion-referencing, and Bloom. In Proceedings of the 34th SIGCSE technical symposium on Computer science education (SIGCSE '03). ACM, New York, NY, USA, 143-147. 2003.

[18] Salman K., Khan Academy. Available: https://www.khanacademy.org/. Last accessed 30th June 2015. 2006.

[19] Hu, D. (2011). How Khan Academy is using Machine Learning to Assess Student Mastery. Available: http://david-hu.com/2011/11/02/how-khan-academy-is-using-machine-learningto-assess-student-mastery.html. Last accessed 30th June 2013.

[20] Schaeffer G. A., Steffen, M., Golub-Smith M. L., Mills C. N., Durso, R. (1995). The introduction and comparability of the computer adaptive GRE general test. Princeton, New Jersey: Educational

Testing Service.

[21] Weiss D. J. (2004). Computerized adaptive testing for effective and efficient measurement in counseling and education. Measurement and Evaluation in Counseling and Development,37, 70-84.

[22] Tom Mitchell, Machine Learning, McGraw Hill, 1997

[23] University of Helsinki, Guidelines for creating learning-objective matrices https://www.cs.helsinki.fi/en/administration/guidelines-creating-learning-objective-matrices. Last accesed 29th November, 2015.