Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 45 (2015) 70 – 75

Procedia Computer Science

International Conference on Advanced Computing Technologies and Applications (ICACTA-2015)

Adaptive Testing and Performance Analysis using Naive Bayes Classifier

Sanjana Agarwal[a], Nirav Jain[b], Surekha Dholay[c]

[a]*Computer Engineering Department, Sardar Patel Institute of Technology, Mumbai 400058, India*
[b]*Computer Engineering Department, Sardar Patel Institute of Technology, Mumbai 400058, India*
[c]*Professor, Computer Engineering Department, Sardar Patel Institute of Technology, Mumbai 400058, India*

Abstract

The highlight of this paper is to demonstrate the concept of adaptive tests which are an efficient way to produce the desired result as compared to the traditional static tests. The algorithm learns to adapt to the user's knowledge, and thus, calculates the true ability of the user. Our test system also provides the user with his performance review in specific categories, thereby giving him an understanding of his current knowledge for the same. Thus, our model can improve the user's efficacy over a prolonged period of time.

Peer-review under responsibility of scientific committee of International Conference on Advanced Computing Technologies and Applications (ICACTA-2015).

Keywords: naïve bayesian; adaptive testing; performance analysis; machine learning; classification; personalized learning; training; performance review; classical test theory; item response theory;

1. Main Text

1.1 Introduction

Our system has been implemented using a Supervised Machine Learning algorithm, Naive Bayes Classifier. A test consists of three sets. Each set of the test is dynamically built with respect to the specific categories of the Language C which will be enumerated upon in the further section of the paper. Naive Bayesian in machine learning is a popular classification algorithm, based on the application of Bayes theorem. It is often used for document categorization, i.e. classifying documents in to one or more categories or classes such as "spam" or "not-spam." Our study requires identification of the difficulty level of all the questions in the next set, depending on the user's performance in the current set. Also, Naive Bayesian is good for variables of few categories. It requires the

Peer-review under responsibility of scientific committee of International Conference on Advanced Computing Technologies and Applications (ICACTA-2015).
doi:10.1016/j.procs.2015.03.088

categories to be independent. Thus, using Naive Bayesian Classifier, we identify whether the next question of a particular category should be classified as “Easy”, “Easy-Medium”, “Medium”, “Medium-Hard” or “Hard.”

Note that we are using Naive Bayesian for two purposes. First type of classification is purely to determine the difficulty level associated with the future questions of a particular category which is done after every set. Thus, the input data consists of the questions asked in that particular set and the output classes are “Easy”, “Easy-Medium”, “Medium”, “Medium-Hard” or “Hard.” Once the user reaches the end of the test (after three sets in our model), we again use Naive Bayesian to determine the user’s knowledge in a particular category. But now, the classification is done only in two classes, namely “Yes” and “No” and the data based on which the classification is done is the overall performance of the user (i.e. all the questions asked in the test.) Using this classification, we estimate how accurately the user can answer the next unseen question of a particular category. Hence, using the overall performance in all the sets presented to the user, we are determining the user’s ability of correctly answering a question in a given category.

1.2 Related Work

Several frameworks exist for developing test systems. One such framework is the Classical Test theory (CTT). It is a body of related psychometric theory that predicts outcomes of psychological testing such as the difficulty of items or the ability of test-takers. In the CTT model, there are three fundamental concepts: test score (sometimes called observed score), true score T, and error score E. In the simple CTT model, the observed test score X of an examinee for a certain test item can be expressed as the sum of the true score and error score. Classical Test Theory has several advantages. The models are straightforward and easy to understand as well as apply in testing practice. Many useful models and methods have been formulated from the CTT framework and have addressed effectively important issues in measurement. The mathematical analysis required for CTT models are usually simple. However, it has several drawbacks as well. It is test oriented, not item oriented. Test oriented means that many CTT models are evaluated based on scores in a test and thus, it is difficult to predict how an examinee will perform on a particular test item. Therefore, CTT models cannot be reliable in predicting an examinee's performance on a specific test item. Adjusting a test to meet the performance level of each candidate has several problems associated to it and may be unfair. It is difficult to compare examinees who take different tests and to compare test items whose characteristics are estimated from different groups of examinees. Also, it is difficult to adjust a test to meet the performance level of each individual candidate. Students generally do not get an insight about their knowledge in particular categories of a subject. Hence, this approach computes the user's knowledge in a given category of language C. This study is important because, depending on the analysis, the user can improvise on the area in which he or she is weak, and thus prolonged use of these tests will help improve the user's knowledge in the given categories. It is interesting as the tests adjust to meet the performance level of each individual candidate. The previous forms of algorithm are test oriented. This is where this system differs, that is, it is item oriented. It consists of an “item bank,” which consists of items selected from a collection of items.

Wikipedia explains that, Item Response Theory (IRT) attempts to model the relationship between an examinee’s latent ability and probability of the examinee correctly responding to a certain test item. Many IRT models exist differing in the mathematical form of item characteristic function and the number of parameters specified in the model namely One-Parameter Logistic Model (1PLM), Two-Parameter Logistic Model (2PLM), and Three-Parameter Logistic Model (3PLM). Past studies, however, have pointed out that the c-parameter of 3PLM should not be interpreted as a guessing parameter. Studies have found logical, empirical evidence showing that none of the three parameters of 3PLM (a-, b-, or c-) can precisely show the discrimination, difficulty, and guessing characteristics of an item, respectively. Hence, we developed a model using a different approach with the help of Naive Bayes Classifier.

1.3 Methodology

The test is objective, wherein, it includes multiple choice questions. It is a timed test, that is, the entire test (consisting of 45 questions) runs for 60 minutes. There will be a one minute break between the three sets of the test.

The subject of the test, that is, language C, is divided into various categories. Five categories (A, B, C, D, and E) form a part of one test, with each test consisting of three sets of 15 questions each. For our model, the five categories are "Declaration and Initialization", "Control Structures", "Arrays and Strings", "Functions and Pointers" and "Structures".

We define a tuple as a set consisting of

- Question
- Category of the question
- Difficulty level of the question
- Correct Answer
- User Answer

And the structure of the 15 questions in a particular set is as follows:

- 5 pure category based questions. For example, a question based on the concept of "Arrays and Pointers".
- 10 mixed category questions. For example, a question based on "Declaration and Initialization" and "Arrays and Strings".

Thus, a particular set will have questions on categories A, B, C, D, E, AB, AC, AD, AE, BC, BD, BE, CD, CE, and DE.

Table 1 displays a sample set of a test (Note that all the questions in the initial set have "Medium" difficulty level). For illustration purpose, we assume that the knowledge of the user in the respective categories is as follows:
A (Declaration and Initialization) - Good
B (Control Structures) – Very Good
C (Arrays and Strings) - Fair
D (Functions and Pointers) - Poor
E (Structures) – Very Poor

Table 1. An example set, displaying the performance of the user as well as the calculated difficulty level in the next set.

Question No	Category	Difficulty Level	Correct Answer Choice	User Answer Choice	Normalized Values	Next Level
1	A	Medium	a	a	1.00	Hard
2	B	Medium	c	c	1.00	Hard
3	C	Medium	d	d	0.67	Medium Hard
4	D	Medium	d	a	0.34	Easy Medium
5	E	Medium	a	b	0.00	Easy
6	AB	Medium	c	c	1.00	Hard
7	AC	Medium	d	d	0.84	Hard
8	AD	Medium	a	b	0.67	Medium Hard
9	AE	Medium	c	d	0.50	Medium
10	BC	Medium	c	d	0.84	Hard
11	BD	Medium	a	a	0.67	Medium Hard
12	BE	Medium	d	a	0.50	Medium
13	CD	Medium	b	c	0.50	Medium
14	CE	Medium	c	a	0.34	Easy Medium
15	DE	Medium	a	b	0.17	Easy

After answering the 15 questions, the difficulty level of the next question of each of the 15 categories is determined using Bayesian Classification using Eq. 1,

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \quad (1)$$

Where,
P (A) = Probability of the occurrence of event A.
P (B) = Probability of the occurrence of event B.
P (A | B) = Probability of A conditioned on B and,
P (B | A) = Probability of B conditioned on A.

With reference to Table 1, let us calculate the Bayesian Value of B after Set 1.The total number of correct answers in this set are 8 and the wrong answers are 7.

Therefore,
P (B | Yes) = [(No. of Correct Answers of B) / (No. Of correct answers)] * P (Correct Answers in the set)
P (B | Yes) = (3 / 8) * (8 / 15) = 0.2
P (B | No) = (2 / 7) * (7 / 15) = 0.13

We consider only the 'Yes' value as we want to check the ability of the user to answer an unseen question correctly. We do this for all pure categories and get their Bayesian values.

Table 2. Bayesian values of pure categories.

Category	Total Number of right answers in a set	Total Number of wrong answers in a set (5 – No. of Right)	Bayesian Values P(Category|Yes)
A	3	2	0.200
B	3	2	0.200
C	2	3	0.130
D	1	4	0.067
E	0	5	0.000

To normalize the Bayesian value for Yes, we use the following Feature Scaling formula:

$$X' = \frac{X - X\min}{X\max - X\min} \quad (2)$$

As seen in table 2, we have Xmax = 0.2 and Xmin = 0.
Therefore, normalized value for category B = 0.2 – 0 / (0.2 - 0) = 1

For a mixed category, the normalized value is computed as the mean of the individual category normalized values. For example, the normalized value of category AB will be:
(Normalized Value of A + Normalized Value of B)/2 = (1 + 1) / 2 = 1

Based on these normalized values and the current difficulty level, we estimate the difficulty level of each type of question in the next set using the following table:

Table 3. Chart to determine the difficulty level of the next question of the category based on the current difficulty level and the normalized values

Easy:	Easy	Easy Medium	Medium
	0.00 – 0.33	0.34 – 0.66	0.67 – 1.00

Easy Medium:	Easy	Easy Medium	Medium	Medium Hard
	0.00 – 0.25	0.26 – 0.50	0.51 – 0.75	0.76 – 1.00

Medium:	Easy	Easy Medium	Medium	Medium Hard	Hard
	0.00 – 0.20	0.21 – 0.40	0.41 – 0.60	0.61 – 0.80	0.81 – 1.00

Medium Hard:	Easy Medium	Medium	Medium Hard	Hard
	0.00 – 0.25	0.26 – 0.50	0.51 – 0.75	0.76 – 1.00

Hard:	Medium	Medium Hard	Hard
	0.00 – 0.33	0.34 – 0.66	0.67 – 1.00

Since the normalized value of B is 1 and the current level is Medium, we get the next difficulty level for category B as "Hard". Note that for this purpose, Bayesian values are normalized.
The same adaptation is done for all category questions, and thus we construct the "Next Level" column as shown in Table 1.

Thus, after the first 15 questions presented to the user, the next 15 adapt dynamically. Following the above mentioned procedure, the test goes on for three sets, and hence, a user answers 45 distinct questions.

At the end of 45 questions, Bayesian Algorithm is used to calculate the user's knowledge in a particular category. A similar calculation is done, but instead of 5 questions being considered, all 15 questions of one category are considered. The normalized values (of the Bayesian values) are multiplied by 100, which gives the percentage of user's knowledge in a particular category as shown in Table 4

Table 4. Knowledge of the user in each of the pure category.

Category	Bayesian Value	Normalized Value	Knowledge
A (Declaration and Initialization)	0.1778	0.7778	77.78 %
B (Control Structures)	0.2222	1.0000	100 %
C (Arrays and Strings)	0.1333	0.5555	55.55 %
D (Functions and Pointers)	0.0889	0.3335	33.35 %
E (Structures)	0.0222	0.0000	0 %

Our results resemble the user's knowledge in respective categories. We worked out a few more examples and got similar results. Since a user's knowledge cannot be computed mathematically, we do not have a metric to verify our answer with. Hence we define our algorithm's accuracy subjectively. We tested our system with a few people, who gave an A grade to our system.

1.4 Conclusion

In this paper we have presented a method and an algorithm to develop a system that can help the user identify his weak areas. The questions are not static, they are dynamically generated after each set. This helps the test to truly adapt to the user's knowledge. And then, at the end of three sets, we give the user his performance analysis in the predefined categories. Based on the review he receives, he can work on his weak areas and give the test again. Thus, an extended usage to such a testing system can help a student gradually improve his knowledge.

1.5 Future Scope

We plan to use the model for various other subjects like Java, Data Structures and Algorithms, Operating Systems, and Database Management to provide training in some of the most important subjects required to work in IT industries. Also, the user can be given an option to save his progress, so that he can keep a track of his performance so far. Thus, it can serve as a personal inductive training test series for the user.

Acknowledgement

As the authors of this paper, we would like to thank our mentor, Mrs. Surekha Dholay for her patient guidance, overwhelming encouragement and useful critiques on this research. We would like to thank our colleagues for their assistance with the evaluation of our system.

We would also like to thank Elsevier for providing a podium to publish our paper. Lastly, we would like to thank the authors of various reference materials mentioned in the references for their commendable research.

References

1. Nathan A. Thompson, Assessment Systems Corporation, David J. Weiss, University of Minnesota "A framework for the development of computerized adaptive tests", Volume 16, Number 1, January 2011
2. Minh Duong, "Introduction to item response theory and its applications", December 2004
3. John Michael .Linacre, "Computer-adaptive testing: A methodology whose time has come." published in Sunhee Chae, Unson Kang, Eunhwa Jeon, and J. M. Linacre. (2000) Development of Computerized Middle School Achievement Test [in Korean]. Seoul, South Korea: Komesa Press.
4. Kyung T. Han, "Fixing the c parameter in the three-parameter logistic model", Volume 17, Number 1, January 2012
5. David Barber c 2007,2008,2009,2010, "Bayesian reasoning and machine learning", DRAFT March 9, 2010