Comparison of Different Machine Learning Algorithms for the Initialization of Student Knowledge Level in a Learner Model-Based Adaptive E-Learning System

ROBERT O. OBOKO, PETER W. WAGACHA, & ELIJAH OMWENGA*
School of Computing and Informatics, University of Nairobi

ARNO LIBOTTON
Faculty of Psychology and Educational Sciences, Free University of Brussels

---

Abstract:
Web-based learning systems give students the freedom to determine what to study based on each individual student's learning goals. These systems support students in constructing their own knowledge for solving problems at hand. However, in the absence of instructors, students often need to be supported as they learn in ways that are tailored to suit a specific student. Adaptive web-based learning systems are suited to such situations. In order for an adaptive learning system to be able to provide learning support, it needs to build a model of each individual student and then to use the attribute values for each student as stored in the student model to determining the kind of learning support that is suitable for each student. Examples of such attributes are student knowledge level, learning styles, student errors committed during learning, the student's program of study, gender and number of programming languages learned by the student of programming. There are two important issues about the use of student models. Firstly, how to initialize the attributes in the student models and secondly, how to update the attribute values of the student model as students interact with the learning system. With regard to initialization of student models, one of the approaches used is to input into a machine learning algorithm attribute values of students who are already using the system and who are similar (hence called neighbors) to the student whose model is being initialized. The algorithm will use these values to predict initial values for the attributes of a new student. Similarity among students is often expressed as the distance from one student to another. This distance is often determined using a heterogeneous function of Euclidean and Overlap measures (HOEM). This paper reports the results of an investigation on how HOEM compares to two different variations of Value Difference Metric (VDM) combined with the Euclidean measure (HVDM) using different numbers of neighbors. An adaptive web-based learning system teaching object oriented programming was used. HOEM was found to be more accurate than the two variations of HVDM.

Categories and Subject Descriptions: H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – User Centered Design; H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia-Navigation, User issues; I.2.6 [**Artificial Intelligence**]: Learning – Concept learning; Induction; K.3.1 [**Computers and Education**]: Computer Uses in Education – Distance Learning, Computer Assisted Instruction (CAI)
General Terms: Algorithms, Human Factors, Experimentation, Measurement
Additional Key Words: Learner modeling, initialization, web-based learning, nearest neighbors, overlap measure, knowledge level, object oriented programming, value difference metric.

IJCIR Reference Format:

---

* Robert O. Oboko, Peter W. Wagacha, & Elijah Omwenga, School of Computing and Informatics, University of Nairobi{roboko, wagacha, muthoni, eomwenga}@uonbi.ac.ke; Arno Libotton, Faculty of Psychology and Educational Sciences, Free University of Brussels, arno.libotton@skynet.be

International Journal of Computing and ICT Research, ISSN 1818-1139 (Print), ISSN 1996-1065 (Online), Vol.3, No.1, pp. 49-56, June 2009.

Robert O. Oboko, Peter W. Wagacha, Elijah Omwenga and Arno Libotton. Comparison of Different Machine Learning Algorithms for the Initialization of Student Knowledge Level in a Learner Model-based Adaptive e-Learning System. International Journal of Computing and ICT Research, ISSN 1818-1139 (Print), ISSN 1996-1065 (Online), Vol.3, No.1, pp 49-56. http://www.ijcir.org/volume3-number1/article6.pdf

---

1. INTRODUCTION

Learner models contain sets of values of attributes of students which identify a specific student. Initialization of student models for web-based learning system involves estimating the level at which a student joins a course with regard to a specific attribute. For instance, if it is a assumed that the course a student is learning is divided into concepts, then initialization may involve estimating the student's knowledge level for a specific concept before the student learns the course and then takes a test at the end. Web-Based systems are those applications or services that are resident on a server that is accessible using a Web browser and is therefore accessible from anywhere in the world via the Web.
Initialization of student models is important because it seems unreasonable to consider every student to be starting up with the same level of knowledge or even having no knowledge at all about the domain being learned e.g. object oriented programming. The system can lose credibility with the students [DeBra, 2000] because of using an inaccurate student model to generate inappropriate advice for students.

In this study, the students' knowledge level for a given concept is initialized i.e. it is estimated before the student takes a test. This estimated knowledge level is used to suggest to the student his/her level of readiness to study a particular concept. This was necessary in order to suggest concepts based on plausible criteria [Virvou and Tsiriga 2004] instead of other measures such as assuming that all students start at the same level of knowledge or no knowledge at all about a particular concept.
In this study, initialization was done using the scores of the students who had already studied the concept under consideration and who were of the same overall knowledge level (novice, intermediate or advanced) as the current student. Both domain independent and indirectly domain dependent features were considered [Virvou and Tsiriga 2004]. Domain independent features are student features whose values do not vary with the domain being studied, such as program of study and gender. Indirectly domain dependent features are those student features whose values do not depend on the domain directly, such as the number of programming languages already learned by the student. The combined vector of feature values for each student together with each student's knowledge score for the current concept were used as training examples for the initialization process. The feature value vector was also formed for the current student for who initialization was being done. This formed the query instance.

To determine which students to consider among the students of the same knowledge level as the current student, only those students who were closest to the current student were considered. They are called nearest neighbors [Mitchell 1997] because if their feature values are plotted on an n-dimensional space, their points would appear closest to the plotted point for the current student. The 1, 3, 5, … nearest students are considered i.e. the k-nearest neighbors.

Some of the student features were numeric and the others were nominal. A numeric attribute is also called a linear attribute. It can be continuous or discrete. A continuous attribute (also called continuously valued) uses real values. Examples of continuous attributes are the weight of a person and the speed of a car. A linear discrete attribute (also called integer attribute) can only take discrete values such as the number of programming languages. Nominal attribute values are not suitable for linear distance measurement numerical attributes are [Wilson & Martinez 1997]. Different attributes can take different types of values, some taking nominal values others numerical values. Hence, a heterogeneous distance function may be used. Different kinds of distance functions are used on different kinds of attributes hence different functions for nominal attributes and different functions for numerical attributes.

Instance-based learning algorithms typically have been used for initializing models of students in intelligent tutoring systems as well as web-based intelligent tutoring systems [Virvou & Tsiriga 2004]. They are machine learning algorithms in which learning occurs by simply storing the presented data. When a new query instance is encountered, a set of similar related instances is retrieved from memory and used to classify the new query instance [Mitchell 1997]. An example of instance-based algorithms is the nearest neighbor algorithm. In this study, the students nearest to the current student are considered and the feature vectors and the knowledge level for each student for a given concept are retrieved for use in the initialization process for the new student.

The instance-based algorithms typically handle numeric attributes well using the Euclidean distance measure. However, they do not handle nominal attributes as well. Mostly, they use the overlap measure for nominal attributes The overlap measure loses some information in the process of comparing the values of two feature vectors [Virvou & Tsiriga 2004; Wilson & Martinez 1997]. The Value Difference Metric (VDM] was designed to find reasonable distances between nominal attribute values.

This paper reports results of investigation on the use of VDM for nominal attributes in the initialization of student models in a web-based intelligent tutoring system. The focus is on accuracy in the prediction of initial student scores for different concepts of an object oriented programming course. Comparison is also made of a heterogeneous function using Euclidean and VDM distance measures with another heterogeneous function using the Euclidean and overlap distance measures.

2. CONTEXT OF THE STUDY

The experiment was run for an object oriented programming course delivered by an adaptive web-based learning system. The second year students at the School of Computing and Informatics of the University of Nairobi were the participants. These students were studying object oriented programming for the first time. They had already studied a programming methodology course which is a pre-requisite for this course.

The students studied the course on their own during laboratory sessions, with minimal support from the course lecturer and the researchers. The experiment was run between February and March 2008. The students started studying the course at different times and the models of those who started later were initialized using the attribute values of those who were ahead of them in studying the course. The attribute of interest in the student model, and which is the subject of the initialization process is the student's estimated knowledge level.

The course was organized into goals. Each goal had one or more concepts, with the content of the concepts of a goal being considered sufficient to achieve the learning goal by the student. In turn, each concept was organized into a number of modules which contained the actual content of the course.

3. REVIEW OF RELATED LITERATURE

Virvou and Tsiriga [2004] used a heterogeneous function with the overlap measure for nominal attributes and the Euclidean distance measure for numerical attributes. The overlap measure compares two values and returns 1 if the values are different and 0 if they are the same [Mitchell 1997]. The Euclidean distance between two training examples (considering all the numerical attributes in the feature vector) is square root of the sum of the square of the difference between pairs of numerical attribute values [Mitchell 1997].

Before learning for a student starts, a set of domain independent attributes e.g. program of study and indirectly domain dependent attributes e.g. number of programming languages already learnt by the student are used to form the initial student model [Virvou and Tsiriga 2004]. The attributes in this model are used to find the distance between two students. These attributes are those considered important to the domain being learned. In this study, these attributes were arrived at after studying related literature [Burkhardt et al 2002; Schmidt 2005; Wiedenbeck 2005; Ventura 2005].

Then the feature vector of the new student is compared to the vectors of already registered students, who have had time to learn using the system. The total distance obtained between two student vectors is used to determine the nearest neighbours for the new student and how the score of each one of them for a given concept will contribute during the estimation of the initial knowledge level of the new student for the same concept. The distance weighted k-NN algorithm is used for the estimation and the weight is the inverse of the distance [Mitchell 1997], meaning that students who are furthest from the new student contribute the least in the initialization of the new student's knowledge level for a given concept.

The overlap measure looks for the equality or otherwise of values, without taking into consideration the class of the training example (i.e. categorization of the student) [Wilson & Martinez 1997] or how close the nominal values are from each other [Virvou & Tsiriga 2004]. For example, the attribute "preference for use of hypermedia during learning" may have possible values as low, moderate and high. The overlap measure would consider the difference between low and moderate as being the same as the difference between low and high just because the values are not the same in both cases. The metric used in this experiment to estimate the distance between two nominal attribute values i.e. VDM takes into consideration not only the similarity of the nominal attribute values but also the

similarity/correlation between the classes of the training examples (i.e. the classes of the two students) being considered [Wilson & Martinez 1997].

With VDM, the distance between two values *x* and *y* of a nominal attribute *a* is defined by the formula:

$$\mathrm{VDM_a}(x, y) = \sum_{c=1}^{C} |(N_{a,x,c} / N_{a,x}) - (N_{a,y,c} / N_{a,y})|^q$$

where $N_{a,x}$ is the number of instances in a training set T

$N_{a,x,c}$ is the number of instances in a training set T that have *x* as the value of attribute *a* and output class *c*

C is the number of output classes in the problem domain

q is a constant, usually 1 or 2.

In the RISE system [Domingoes 1995], q was equal to 1. On the other hand, in the empirical study [Wilson & Maritinez 1997], values of both 1 and 2 were used for q. The experiment was conducted over 15 databases from the machine learning database repository of the University of California, Irvine, all with nominal- and numeric-valued attributes and because of this, a heterogeneous distance function was required. Euclidean and VDM distance measures were used. Q = 2 was found to be more robust that Q = 1.

4. METHODOLOGY

There were two aims of the experiment: To evaluate the effectiveness of student model initialization and to compare VDM with the overlap measure.

The students took an adaptive pre-test for each concept. The pre-test was adaptive so that students did not have to take all questions in the pre-test for a given concept, especially if they were starting to learn object oriented programming [Yau 2004]. The overall pre-test score across all the concepts was used to place a new student into one of novice, intermediate and advanced classes.

Then the heterogeneous distance functions (HOEM and VDM) and distance-weighted k-NN algorithm were used to estimate the initial knowledge level for each student, based on the models of the other students of the same class who had interacted with the system for some time. Different numbers of neighbors were also considered for the experiment. Where there were no other students already in a givens class, the pre-test scores of the student were used as the student's initial scores.

The initialization scores were expressed approximately to the possible pre-test scores (0, 4, 8, 13, 18, 24 and 30) and therefore, it was possible to compare them to see if they were similar to the pre-test scores for any given student and for any given concept was possible.

The scores initialized using HVDM with q = 2 were used for proposing the concepts the system considered the student ready to study. This is because this variation of HVDM had been found to be more robust than HVDM with q=1 [Wilson & Maritinez 1997].

5. COMPARISONS WITH RELATED WORK

In the work of Virvou and Tsiriga [2004], the initialization of student models is based on default assumptions, which are attached to each class, called a stereotype. This means that students of the class will have common initial knowledge level scores. They have not given the students a pre-test because a student might score in a pre-test as a result of a 'simple guess or slip', leading to reduction in the accuracy of the student model. pp 292.

In this experiment, we provided pre-tests for each concept. Each pre-test had six questions. There were two pre-test questions per level (novice, intermediate, advanced) instead of one question so as to reduce chances of a 'slip' or 'guess'. Answers to both questions were used to determine a student's class. If both were answered correctly, then the student was provided with questions of the next level. If one of the questions was answered incorrectly, the student got another chance to answer it. But if both were wrong (student was in the previous level), or the second question was wrong on second attempt (student was in current class), then the class was confirmed and no more questions were provided (adaptive pre-testing). This reduced the time spent in the pre-test, an issue raised by Virvou and Tsiriga [2004], but at the same time leading to a reasonable estimation of the student's initial knowledge [Guzman & Conejo 2002].

In this study, instead of using default scores (when the student first registered his/her category), the student's pre-test scores were used for initialization since they were already individualized and were real scores. When there were other students already of the same class who had

interacted with the learning system for some time and their initial scores had been updated, then their scores were subjected to the distance-weighted k-NN to estimate the initial concept scores for the new student.

In the same study of Virvou and Tsiriga[2004], the heterogeneous distance function used had the Euclidean measure for numeric attributes and the overlap measure for nominal attributes. The overlap measure only compares the similarity of the values without considering the class of the training example, yet this information can be used for better estimation of distances between feature vector values as happens with VDM [Wilson & Maritinez 1997]. In our study, VDM was used.

6. RESULTS OF THE EXPERIMENT

The results reported in this paper are part of an ongoing experiment at the School of Computing and Informatics, started in February 2008 and running into March 2008. Three different machine learning algorithms were compared in terms of their accuracy at initializing the concept knowledge scores of a student, based on the scores obtained by similar students (neighbours) who were of the same knowledge category. The knowledge categories were Novice, Intermediate and Advanced. The three different machine learning algorithms used were HOEM, HVDM with the value Q=1 and HVDM with the value Q=2.

The first part of the experiment, whose results are being reported in this paper, had 8 concepts which students were supposed to learn. Each concept had a pre-test of six questions. The six questions were organized into three categories namely Novice, Intermediate and Advanced. Each category had two questions. The Advanced level questions had a score of 6 each, the questions of the Intermediate level had a score of 5 each and the questions of the Novice level had a score of 4 each. Therefore, the possible scores a student could obtain in a test were as in the Table 1 below:

Scenario	**Level1 Questions Right**	**Level2 Questions Right**	**Level3 Questions Right**	**Level1 Score**	**Level2 Score**	**Level3 Score**	**Total Score**
1	0	0	0	0	0	0	**0**
2	1	0	0	4	0	0	**4**
3	2	0	0	8	0	0	**8**
4	2	1	0	8	5	0	**13**
5	2	2	0	8	10	0	**18**
6	2	2	1	8	10	6	**24**
7	2	2	2	8	10	12	**30**

Table 1: Possible scores in a pre-test for a student

Note: There were no such scores as 1, 2, 3, 5, 15 or 27. Scores obtained by a student were always one of the 7 possible scores i.e. 0, 4, 8, 13, 18, 24 or 30. Scores initialized were also rounded to be one of the 7. Scores were considered close to each other if they were next to each other in the order of the 7 possible scores given above. For example, 4 is close to 0 and 8, 13 is close to both 8 and 18.

The accuracy of an algorithm was obtained by comparing the score obtained by the student in the pre-test and the initialized score of the student for the concept according to the algorithm (HOEM, HVDM with Q=1 or HVDM with Q=2) and the number of neighbours considered during the initialization process (1, 3, 5, 7, … i.e. odd numbers).

Two measures of accuracy were considered. One of the measures considered the exactness of the two sets of scores i.e. those initialized and those obtained by the student in the pre-tests for the different concepts. Therefore 4 and 4 are considered the exact and 0 and 4 are considered completely different.

The exactness is indicated by an entry of 1 in the column of exactness in Table 2 below. Inexactness of the scores was indicated by an entry of 0 in the exactness column in Table 2.

The second measure expressed the goodness of the classifier's estimation of the initial scores for the student. If an initialized score was either exact or next to the pretest score such as 4 to 8 and 0, then the estimation was considered good. This is indicated by an entry of 1 in the column of goodness of the estimate in Table 2 below. Otherwise, the comparison was considered not good and an entry of 0 was made in the 'goodness' column in Table 2.

Table 2 below shows a segment of the summary of the comparison of the initialized scores with their corresponding pre-test scores using HOEM. There were similar summaries for HVDM with Q=1and HVDM with Q=2.

			Number of Neighbours											
			1			3			5			7		
Stud No	**Concept**	**Pretest Score**		**Exact Estimate**	**Good Estimate**		**Exact Estimate**	**Good Estimate**		**Exact Estimate**	**Good Estimate**		**Exact Estimate**	**Good Estimate**
10	**G1_1**	**4**	0	0	1	0	0	1	0	**0**	**1**	0	**0**	**1**
	G2_1	**24**	24	1	1	24	1	1	**24**	**1**	**1**	**24**	**1**	**1**
	G2_2	**13**	18	0	1	**13**	**0**	1	13	**1**	**1**	13	**1**	**1**
	G2_3	**24**	8	0	0	8	0	0	8	**0**	**0**	8	**0**	**0**
	G2_4	**0**	8	0	0	8	0	0	8	**0**	**0**	4	**0**	**1**
	G3_1													
		Total	**5**	**1**	**3**		**1**	**3**		**2**	**3**		**2**	**4**
		Percent	**%**	**20**	**60**		**20**	**60**		**40**	**60**		**40**	**80**

Table 2: Tabulation of 'Exact' and 'Good' comparison of initialized scores using HOEM with different numbers of neighbours.

There was a summary foe each of the three different algorithms, similar to the results shown in Table 2 above. From the three summaries, it was possible to compare the three different algorithms. The 'good' results for 1 nearest neighbour and the best result obtained when other different numbers of neighbours were considered are tabulated in Table 3 below:

6. DISCUSSION

Metrics for expressing the distance between two neighbours, especially as used with the k-nearest neighbour algorithm, are many. It is has been proved using different datasets that HOEM does not do as well as HVDM with both Q=1 and Q=2[Wilson & Maritinez 1997]. The research being reported on in this paper was carried to see if this proof applied to data about students learning a course using an adaptive web-based course. Other researchers who have used HOEM in adaptive web-based learning environments [Virvou & Tsiriga 2004] believe that other algorithms for estimating distance between students could do better in initializing student models than HOEM. This was because HOEM only considered the difference between two nominal attribute values and did not consider the class of the entities described by the nominal attributes.

From the results obtained from this research and presented in Table 3, HOEM seems to be doing better than HVDM with Q=1 and Q=2 because it has on average attained higher 'good' classification accuracy. However, it is important to note that only 17 students used adaptive support to learn the first 8 concepts of the course offered through our adaptive web-based learning system. The other students who participated served as the control group and did not use adaptive learning support. Had there been more students learning using the system, perhaps studying a full course, the results might have been different. By the end of the experiment, we expect to have more information that can be used to compare the accuracy of initialization of student models using the three algorithms. HVDM is expected to perform better, especially in the exact comparison of initialized and pre-test scores.

HVDM with Q=1 has also attained a higher average accuracy than HVDM with Q=2. This, however, is expected to change [Wilson & Maritinez 1997], especially in the exact comparison of scores.

	Number of Neighbours					
	K=1 (1 Neighbour)			**Best results from other K (other numbers of neighbours)**		
Student Number	**HOEM (%)**	**HVDM1 (%)**	**HVDM2 (%)**	**HOEM (%)**	**HVDM1 (%)**	**HVDM2 (%)**
10	60	60	60	80	80	
13	25	25	25			
14	17	17	13			
16	29	29	29	43	29	29
17	53	57	57			
18	100	100	100			
19	50	43	43	17	43	43
21	25	0	0	0	0	0
24	50	100	75	25	100	75
25	60	80	60	60	80	60
26	29	29	29			
27	67	29	43	83	43	43
28	60	40	60			
Count	**13**	**13**	**13**	**7**	**7**	**6**
Sum	**625**	**609**	**594**	**308**	**375**	**250**
Mean	**48.1**	**46.8**	**45.7**	**44**	**53.6**	**41.7**
Max. Value	**100**	**100**	**100**	**83**	**100**	**75**
Min. Value	**17**	**0**	**0**	**0**	**0**	**0**

Table 3: Comparison of 'Good' estimates of classifier accuracy of the three different algorithms HOEM, HVDM with Q=1 and HVDM with Q = 2.

When initialization was done using more than 1 neighbour, the average accuracy seemed to improve as the number of neighbours increased. This can be confirmed from the example shown in Table 2 on the initialization of concept scores for student 10. The second phase of learning using the system is going on and the students who started using adaptive learning support in this phase are expected to have a higher number of neighbours participating in the concept score initialization process.

7. CONCLUSION

E-learning courses that have small numbers of students may not gain from the expected benefits of using adaptivity. In our example, there were 43 students out of which 17 used adaptivity. For the different scenarios in the study, accuracy was below 54 percent. Large groups of students can provide enough data for the underlying machine learning algorithms to learn and produce better predictions. This is because in the experiment, as the number of neighbours increased, the accuracy of the results became better (see Table 2). Therefore, the online classes with large classes stand a better hence of gaining from the use of machine learning algorithms for classification and prediction.

In the comparison of the three algorithms, HOEM did perform better. Nonetheless, the results among the algorithms were close to each other. Since HOEM has been used for learning systems to do initialization of learner models [Virvou and Tsiriga 2004], it should be used in such systems until further experiments show that the other algorithms perform better.

8. FUTURE WORK

In order to test the algorithms further, experiments should be carried out using a bigger sample than has been the case in this study, where only 43 students participated. This will make it possible for the researchers to monitor better the trends in accuracy as the number of neighbours increases.

The classifiers also need to be tested in a learning environment that is arts based so that the performance of the classifiers can be compared to that of science-based courses such as object oriented programming, which is the course students have studied during this research.

9. REFERENCES

BURKHARDT, J., DE´ TIENNE, F. AND WIEDENBECK, S. 2002. Object-Oriented Program Comprehension: Effect of Expertise, Task and Phase. Empirical Software Engineering, 7, 115–156, 2002

DE BRA, P. 2000. Pros and cons of adaptive hypermedia in web-based education. Journal of Cyberpsychology and Behavior 3(1), 71-77.

DOMINGOS, P. 1995. Rule induction and instance-based learning: A unified approach. In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, Montreal, Canada, pp. 1226 1232.

GUZMAN, E. AND CONEJO, R. 2002. Simultaneous evaluation of multiple topics in SIETTE. In: S. A. Cerri, G. Gouarderes and F. Paraguacu (eds.). In Proceedings of the Sixth International Conference on Intelligent Tutoring Systems, Lecture Notes in Computer Science, Vol. 2363. Springer-Verlag, Berlin, Heidelberg, pp. 739-748

MITCHELL, T. 1997. Machine Learning. McGraw-Hill, New York.

OBOKO, R, WAGACHA, P, LIBBOTON, A., OMWENGA, E. & ODOTTE, Z. 2007. Non-obstrusive determination of Learning Styles in Adaptive Web-based Learning. In Proceedinggs of the Moi University 3rd Annual International Conference. Eldoret, Kenya, pp 157-164

SCHMIDT. C. 2005. Cognitively Flexible Hypertext in an Object Oriented Course: Effects of Case-based Instructional Support on Student Learning. Unpublished thesis, Wichita state University

VENTURA, R. P. 2005. Identifying Predictors of Success for an Objects-First CS1. Computer Science Education Vol. 15, No. 3, September 2005, pp. 223 – 243

VIRVOU, M. AND TSIRIGA, V. 2004. A Framework for the Initialization of Student Models in Web-based Intelligent Tutoring Systems. User Modeling and User-Adapted Interaction 14: 289 - 316, 2004

WIEDENBECK, S. 2005. Factors affecting the success of non-majors in learning to program. In Proceedings of the 2005 international workshop on Computing education research Seattle, WA, USA Pages: 13 - 24

WILSON, D.R. & MARTINEZ, T.R..1997. Improved Heterogenous Distance Functions. Journal of Artificial Intelligence Research 6 (1997) 1-34

YAU, J. Y. 2004. Learning Objects and the Teaching of Java Programming. Unpublished thesis, University of Warwick