

ORIGINAL RESEARCH

Identifying student group profiles for diagnostic feedback using snap-drift modal learning neural network

Samson Habte*, Dominic Palmer-Brown, Miao Kang, Fang Fang Cai

School of Computing, London Metropolitan University, London, United Kingdom

Received: October 10, 2015

Accepted: December 9, 2015

Online Published: December 23, 2015

DOI: 10.5430/air.v5n2p1

URL: <http://dx.doi.org/10.5430/air.v5n2p1>

ABSTRACT

The aim of this paper is to propose a novel method for identifying student group profiles based on student responses to a set of multiple choice questions for the purpose of constructing diagnostic feedback using snap-drift modal learning neural network. The proposed method is capable of supporting tutors without the knowledge of machine learning in identifying useful student groups and constructing diagnostic feedback. Trials were conducted and analysis of the result showed that the snap-drift modal learning neural network was able to identify distinct student groups and represented student group profiles were helpful in revealing gaps of understanding and misconceptions that facilitate construction of diagnostic feedback. Moreover, the result showed that all student responses gathered were assigned to their appropriate student group profiles and the diagnostic feedback constructed based on the identified student group profiles had a positive impact on improving the learning performance of the students.

Key Words: Modal learning, Unsupervised learning, Snap, Drift, Diagnostic feedback, Formative assessment

1. INTRODUCTION

Information and communication technologies have been playing a huge role in improving the efficiency and effectiveness of learning and teaching in higher education. Two decades ago, research studies were focused on how to use artificial intelligence techniques to imitate teachers or tutors in delivering learning sessions. A system that apply artificial intelligence to imitate teachers or tutors is called intelligent tutoring system (ITS).^[1] Most intelligent tutoring systems have four major components: knowledge domain, student model, teaching strategies and user interface.^[2] Student model is the most essential component of any ITS which enables personalized and adaptive learning. There are three approaches to construct student models.^[3] The first approach uses a specially prepared task-model pairings. The second

approach constructs a student model by mapping behaviour to predefined set of bugs. The third approach constructs a student model by inferring the model from observed behaviour. The first two approaches depend heavily on catalogues of mal-rules, which is a simple perturbation of some correct rules collected through an extensive protocol analysis of the domain. In the third approach, the idea is to use a smaller amount of initial knowledge to infer a student model and it is possible by applying machine learning techniques. Machine learning techniques have been applied in several research studies to construct student models in the context of intelligent tutoring systems.^[4-7]

The use of intelligent tutoring systems has been very limited in higher educations as most educational institutions

*Correspondence: Samson Habte; Email: s.habte@londonmet.ac.uk; Address: School of Computing, London Metropolitan University, London, United Kingdom.

are in favour of using virtual learning environments (VLEs), which provide a set of software tools to support learning and teaching. Examples of VLEs are Moodle, WebCT and Blackboard. According to the survey conducted by Universities and Colleges Information Associations (Ucisa), 34% of all higher education institutions in the UK used Blackboard in 2001 and the usage increased to 60% by 2012.^[8] The survey also indicated that “Blackboard is the most used enterprise or institutional VLE, but Moodle has increased in usage as an enterprise solution and remains the most commonly used VLE platform when departmental/school implementations are also considered”. The main focus of VLEs is to support and facilitate teaching and learning, whereas intelligent tutoring systems aim to emulate teachers or tutors in delivering learning lessons. Previous intelligent tutoring systems were focused only on acquiring well defined procedural skills, but did not address helping students developing conceptual understanding.^[9]

Several computer-based assessments, which can be deployed as part of VLEs or independently, have been proposed that support different assessment types such as multiple choice questions,^[10-12] short-free-text responses^[13-15] and problem solving exercises.^[16-19] A literature review on the feedback mechanisms of existing computer-based formative assessment revealed that they had never applied machine learning techniques or student modelling activity and instead only provide item-based feedback mechanism, which is a feedback tied to individual question or feature. Even though machine learning techniques were applied to construct student models as part of intelligent tutoring systems, the student model has never been used to facilitate construction of feedback in the context of formative assessments whose purpose is to provide feedback to improve student learning experiences. Since the student model represents knowledge levels as categories such as beginner, intermediate and advanced, it is not suitable to facilitate generation of diagnostic feedback. To improve the feedback mechanism of computer-based formative assessments, an unsupervised learning, which is snap-drift modal learning neural network has been applied previously in Ref.^[20,21] These previous research studies investigated the application of snap-drift modal learning neural network for analysing student responses to multiple choice questions (MCQs) in order to identify student groups that facilitate generation of diagnostic feedback. The research studies showed that snap-drift modal learning neural network can identify student groups that can represent different knowledge levels based on student responses gathered from MCQs based assessment sessions. The research studies also demonstrated that a diagnostic feedback, which should be constructed based on a combinations of responses not tied to a particu-

lar question, can improve student learning performance in understanding concepts.

This research builds on the previous research studies and its aim is to propose a novel method for identifying student group profiles based on student responses to a set of multiple choice questions for the purpose of constructing diagnostic feedback using snap-drift modal learning neural network. To achieve this aim, firstly, we defined a learning task that needs to be performed by the snap-drift modal learning network. Once the learning task is defined, a snap-drift learning algorithm was implemented using Java and then training data sets were prepared using two real assessment tasks. Secondly, we proposed a method for representing identified student groups and analysed the learning behaviour of the snap-drift modal learning neural networks. Based on the insight gained from the analysis, we defined criteria for determining the usefulness of student group profiles and identified student group profiles using snap-drift modal learning neural networks. Once a set of represented student group profiles were identified for the two assessment tasks, a diagnostic feedback was constructed for each student group based on a recommended guideline. Thirdly, we conducted assessment sessions to gather student responses. Finally, we analysed the student responses gathered to test if student responses are assigned to their appropriate student groups, to assess if student group profiles facilitate the process of identifying gaps of understanding and misconceptions, and to assess the impact of diagnostic feedback, which is generated based on student group profiles, on student learning performance.

The remainder of this paper is organised as follows. In section two, snap-drift modal learning neural network is described. Section three outlines a learning task that defines the problem to be solved using the unsupervised snap-drift modal learning neural network. In section four, the implementation details of the snap-drift modal learning neural network are discussed. Section five explains how training data sets are prepared. In section six and seven, the development of student group profiles and construction of diagnostic feedback are described respectively. In section eight, trials conducted to gather student responses from assessment sessions are discussed. Section nine presents results and discussions. Finally, conclusion and future work are discussed in section ten.

2. SNAP-DRIFT MODAL LEARNING NEURAL NETWORKS

Modal learning neural network combines several modes of learning within a single neural network or module in order to achieve learning results that no single mode could achieve

through exploitation of the complementary nature of each mode.^[20] It is different from hybrid or modular neural networks in which the different learning modes are applied at different modules and/or at separate times. Snap-drift neural network (SDNN) and adaptive function neural network are examples of modal learning neural networks. In adaptive function neural networks, both weight vectors and shape of activation functions are updated simultaneously.^[23]

SDNN is a simple modal learning method, which swaps periodically between snap and drift learning modes. SDNN was first conceived as a learning algorithm as an attempt to overcome the limitations of adaptive resonance theory (ART) learning in non-stationary environments where self-organization needs to take account of periodic or occasional performance feedback.^[24] Snap is a logical intersection learning while drift is a learning vector quantisation. They provide complementary features. Snap captures common elements of group of patterns represented by the minimum values on each input pattern and it contributes to rapid convergence whereas drift captures the average values of the group of patterns.^[20]

The architecture of SDNN consists of three layers.^[20] As shown in Figure 1, they are an input layer, a distributed d layer for feature extraction and a selection s layer for feature classification. The distributed d layer groups the input patterns according to their features using snap-drift training algorithm. The D most activated (winning) nodes or features out of the d layer whose weight vectors best match the current input pattern are used as the input data to the selection s layer. In the s layer, a quality assurance threshold is applied. If the net input of the most active s node is above the threshold, then s node is accepted as the winner and defines the category of the input pattern; otherwise a new uncommitted output node is recruited as the winner.

The weight vectors of the D most activated (winning) nodes and the winner of the s layer are updated according to the following equations.

$$\text{Snap drift} = \alpha(\text{Snap}) + (1 - \alpha)(\text{drift}) \quad (1)$$

The above equation defines how the snap and drift learning modes are combined in SDNNs. In successive learning epochs, the learning is toggled between snap and drift learning modes. When α is set to one, a snap learning is invoked whereas a drift learning mode is invoked when it is set to zero. The above weight update equation is further elaborated in the following equation.

$$W_{ji}^{(new)} = \alpha(I \cap W_{ji}^{(old)}) + (1 - \alpha)[W_{ji}^{(old)} + \beta(I - W_{ji}^{(old)})] \quad (2)$$

Where w_{ji} is a weight vector of either d or s layers, I is a binary input vector, and β is the drift learning rate. After each weight update, the weight vectors are normalized to a unit length.

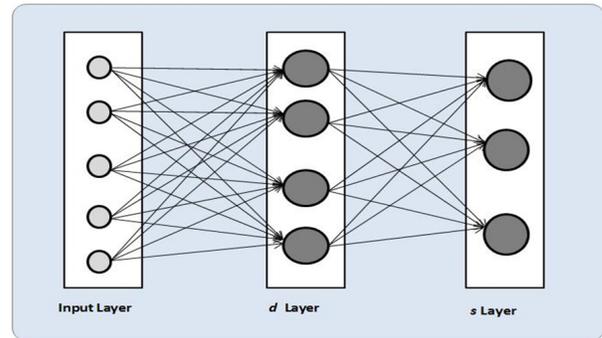


Figure 1. Architecture of SDNN^[20]

3. LEARNING TASK

There are different forms of assessment, which can be used to gather student responses. MCQs is an assessment type that has been introduced in higher education due to its suitability in current higher education environments with a large number of students and reduced resources,^[25] and it can be used to assess different cognitive levels. In addition to this, MCQs can be designed with a diagnostic end in mind, in order to find out whether specific areas of a given subject are adequately known or understood, or in order to detect misconceptions.^[26]

It is not difficult if we only need to generate a feedback tied to an individual question, as feedback can be constructed for each option of a given question. However, this type of feedback is not effective since we can not assess the ability of students to understand a topic or sub-topic using only one question. We can only assess student's ability to recall a particular fact using one question, but we obviously need more than one question to assess student's ability to understand a particular topic. This implies that we should consider different combinations of responses to a set of MCQs to construct effective feedback. For example, if we consider five multiple choice questions with five options (A, B, C, D, E), the total possible number of combinations of responses is 7,750. The total possible number of combinations of responses increases exponentially as the number of multiple choice questions increases. Therefore, it is not feasible to construct a feedback for each combination of responses. Even though each

student is unique, there exists only a limited number of different ways of understanding and range of misconceptions of a topic.^[27] Hence, we only need to identify the different groups which are characterised by similar gaps of understanding, level of understanding and/or common misconceptions and assign a given student responses to an appropriate group automatically.

One possible solution is to analyse manually a collection of responses of students by human expert in order to categorize the responses into groups of similar level of knowledge or understanding. Once the groups are identified each group can then be analysed to characterise its understanding or knowledge level. This solution is very time consuming and is not practically possible for a large number of student responses. Another approach is to use neural networks, which are able to derive meaning from complicated and/or imprecise data and to extract patterns that are too complex to be noticed by many other computational techniques.^[28] These characteristics make neural networks a powerful method to model human behavior and a useful technique to create user models for hypermedia applications.^[28] A neural computing technique, which is snap-drift modal learning neural network is applied to identify the different groups which are characterised by similar gaps of understanding, level of understanding and/or common misconceptions and to assign a given student response to an appropriate group automatically.

The input patterns for the learning task are a set of student responses gathered from assessment sessions. An input pattern (input vector), which is a sequence of responses, is defined mathematically as follows:

$$X = \{x_1, x_2, x_3, \dots, x_n\} \quad (3)$$

Where n is the number of multiple choice questions and

$$x_i = \{a, b, c, d, e\} \quad (4)$$

4. IMPLEMENTATION OF SNAPDRIFT MODAL LEARNING NEURAL NETWORK

The main component of the snap-drift modal learning neural network is the snap-drift learning agent that represents the topology, learning parameters, training patterns, group allocation map and learning algorithms of snap and drift as outlined in section 2. It has one operation that performs learning for a given epoch and returns a group allocation map that contains the index of all training patterns and their corresponding winning nodes of s layer. The algorithm of the operation is described using a pseudo-code as follows:

- (1) Get learning parameters (d, s, D , drift learning rates for distributed and selection layers and quality assurance threshold as explained in section 2).
- (2) Get training data.
- (3) Get current epoch.
- (4) Create an empty group allocation that maps patterns to winning nodes.
- (5) Create d and s layers based on the learning parameters and dimension of training patterns.
- (6) Determine whether the epoch is odd or even and set the learning mode to snap or drift.
- (7) FOR each input pattern of the training data
 - A. Find the D winning nodes at d layer with the largest net inputs.
 - B. Use equation 2 to update the weight vectors of the D winning nodes.
 - C. Normalise the updated weight vectors.
 - D. Set the output of the D winning nodes at d layer to 1 and the output of the remaining nodes to 0.
 - E. Consider the outputs of the nodes in the d layer as input patterns to the s layer.
 - F. Find a node at the s layer with the largest net input.
 - G. IF net input of the node with the largest net input is greater than the quality assurance threshold THEN
 - i. Use equation 2 to update the weight vector of the winning node and set the winning node as committed.
 - ii. Normalise the weight vector of the winning node.
 - iii. Add the current input pattern and the winning node to the group allocation map.
 - H. ELSE
 - i. Select uncommitted node from the nodes of the s layer.
 - ii. Use equation 2 to update its weight vector and set it as committed.
 - iii. Normalise the weight vector of the selected node.
 - iv. Add the current input pattern and the selected node to the group allocation map.
 - I. END IF
- (8) END FOR
- (9) Return group allocation map.

The complete algorithm for the snap-drift modal learning neural network is described using flowchart in Figure 2.

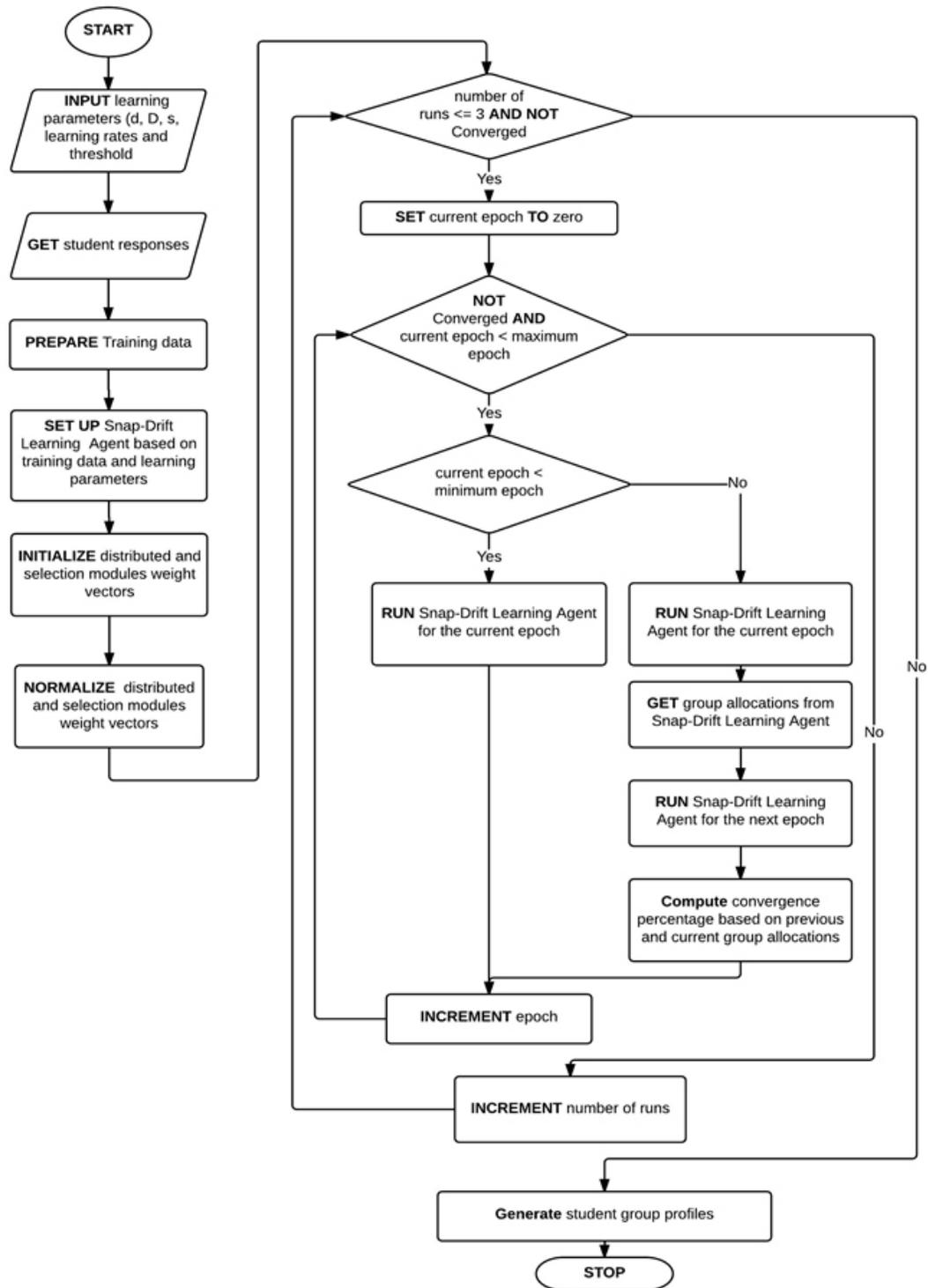


Figure 2. Flow chart of Snap-drift modal neural network algorithm

The Snap-Drift modal learning neural network is implemented in Java. To test whether the learning algorithm is implemented correctly and assess its performance in classifying linearly inseparable patterns, iris data set was used. The result of an experiment that involves ten runs showed

that the minimum number of groups identified was 2 and the maximum number was 9. In all runs, training patterns that belong to the first class were not mixed with the training patterns of the second and third classes. This demonstrates that SDNN learning algorithm was implemented correctly

and was capable of classifying linearly separable data. Training patterns of the second class were also separated from the training patterns of the third class with a small number of patterns mixed from both classes in all other groups that do not contain training patterns from the first class. The average performance for classifying the non-linearly separable classes, which are the second and third classes, was above 95 percent.

5. DATA PREPARATION

Two assessment tasks were chosen to gather student responses. The first assessment task was selected from Introduction to Data Analysis module, which is a core module for first year students at London Metropolitan University. 501 student responses were captured and saved from a web learn where students were allowed to practice multiple choice questions. Five related MCQs that can assess the ability of students in understanding probability topic were considered. The second assessment task was selected from Introduction to Programming module, which is also a core module for first year students at London Metropolitan University. The assessment task was a set of five multiple choice questions on the topic of constructors in the context of object-oriented programming. Paper-based assessment session was undertaken by students who were registered for the module and 115 student responses were captured. In both assessment tasks, the session was carried out in an environment where students were not allowed to help each other. Captured student responses were processed and saved as text file with predefined format where each student attempt is represented by a sequence of characters. The number of characters corresponds to the number of multiple choice questions. The distribution of the student responses captured from the first and second assessment tasks are shown in Figures 3 and 4 respectively.

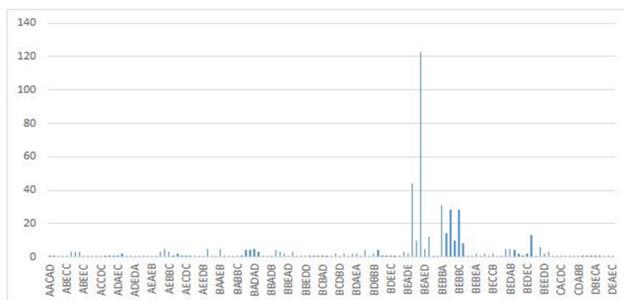


Figure 3. Histogram of the distribution of student responses captured from assessment task one

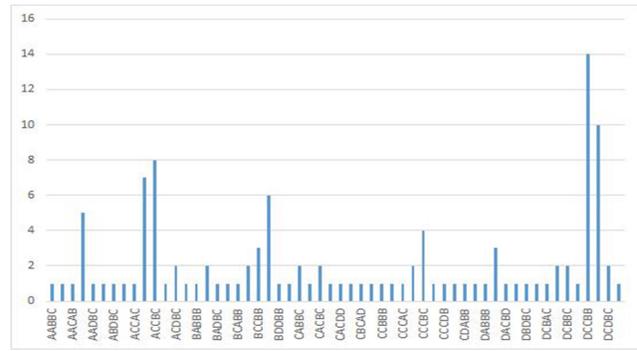


Figure 4. Histogram of the distribution of student responses captured from assessment task two

The sequence of characters that represent captured student responses were transformed into a binary vector using a coding of information, which is popular in information theory field. Based on this coding scheme, each student answer (character) is represented by a binary number with a number of digits equal to the number of options of a multiple choice question. The position of 1 is changed to represent all possible answers. For example, to represent a sequence of answers with four options, we use four digits binary numbers: 0001, 0010, 0100, 1000 to represent A, B, C, D respectively.

6. DEVELOPING STUDENT GROUP PROFILES

The output of the Snap-drift learning agent consists of a group allocation map, a set of winning nodes of *s* layer and two matrices of weight vectors corresponding to the distributed and selection modules. The set of winning nodes represent the different student groups while the two matrices of weight vectors can be used to identify the appropriate student group of a new student response. The group allocation map contains information about the allocation of each training pattern to its appropriate student group. As mentioned previously, the purpose of identifying student groups is to help tutors in revealing gaps of understanding and misconceptions so that they can write an appropriate diagnostic feedback that improves student learning performance. Therefore, it is necessary to represent the group allocation map in a format that facilitates achieving this purpose.

The proposed method to construct student group profile based on the group allocation map is described using a pseudo-code as follows.

- (1) Set group profile threshold that determines the most likely answer.
- (2) Get group allocation map from the Snap-Drift learning agent.
- (3) Create an empty matrix for each student group.

- (4) Set the row and column of each matrix to the number of multiple choice questions and the number of options respectively.
- (5) Compute the value of each element (x_{ij}) of all matrix as the percentage of student responses for question i who answered option j based on the group allocation map.
- (6) FOR each matrix
 - A. FOR each row
 - i. Find the column index with the highest percentage.
 - ii. IF the highest percentage is greater than the threshold THEN Replace the row with an appropriate character (1 = A; 2 = B; 3 = C; 4 = D; 5 = E) corresponding to the column index.
 - iii. ELSE Replace the row with an asterisk (*).
 - iv. END IF

B. END FOR

- (7) END FOR
- (8) Return matrices.

The implemented Snap-Drift modal learning neural network could generate different set of student group profiles based on the values of the learning parameters. To get insight into the relationship among the learning parameters and how they affect the output, several combinations of the learning parameters were tested using iris data set and the training patterns prepared in the previous section. A graphical user interface was integrated with the implemented snap-drift modal learning neural network to assist the testing. As shown in Figure 5, the graphical user interface comprises four components: training data panel, SDNN parameters panel, control granularity of student group panel, training panel and output visualisation panel. The output visualisation panel component is integrated with MATLAB.

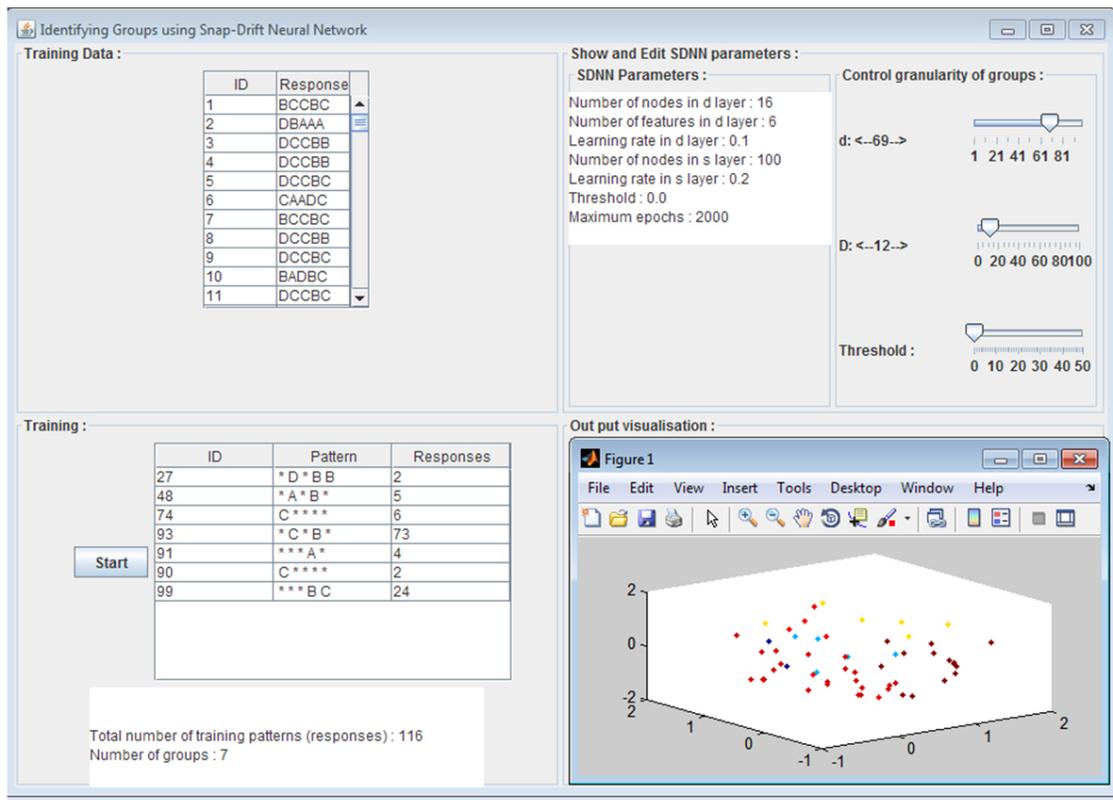


Figure 5. Screen shot of Graphical user interface of SDNN tool

Based on the observation of the tests, the number of nodes in d layer combined with its number of simultaneously active nodes (D) and quality assurance threshold (hurdle) determine the number and nature of groups. Generally, increasing the number of features (D) in d layer decreases the number

of groups. For a given number of features, increasing the quality assurance threshold tends to increase the number of groups. The quality assurance threshold can be used to fine tune the nature of the identified groups. The influence of the threshold starts after a certain value which is correlated to the

number of features in d layer. When its value is greater than a certain value, it becomes hard to identify groups due to slow convergence speed and lack of available uncommitted nodes in s layer. The learning rates for d and s layers influence the convergence speed. The recommended values of the learning rates for d and s layers are 0.1 and 0.2 respectively. Convergence is not guaranteed. If there is no convergence, the trivial solution is to re-start the training process or increase the quality assurance threshold by a small amount.

Generally, too many student groups consume too much tutors' time in writing feedback. On the other hand, a very small number of student groups could be less effective since the feedback might be too generic. Therefore, it is very important to determine the optimal number of student groups in order to construct effective diagnostic feedback. As long as the training data set is large and representative of the potential patterns, the implemented SDNN is capable of converging to an appropriate number of student groups. However, the implemented SDNN might find a small or large number of student groups because of less representative and inadequate training patterns. In this case, we can adjust the value of D , d and quality assurance threshold to decrease or increase the identified student groups.

Based on the method described in the above, a student group is represented by a sequence of most likely responses for each multiple choice question. The method specifies a threshold for determining the most likely responses. Its value should be between 70 and 80. A value below 70 decreases the number of student responses assigned to their appropriate student group, whereas a value above 80 decreases the probability of finding most likely sequence of responses. Even though the group profile threshold is set between 70 and 80, there might be a mix of responses to a particular question. This case happens when percentages of student responses for a given question is evenly distributed over all possible options.

If a student group is represented by a sequence of asterisk (*), which means a mix of responses for all multiple choice questions, it is not a useful student group as there are no most likely responses that reveal any gaps of understanding and misconceptions of a particular topic. The question is what should be the nature or pattern of the student group profile in order to be useful. Since one multiple choice question can not assess any concept or aspect of a given topic, a student group profile that consist of only one most likely response can not be used in revealing any gaps of understanding and misconceptions of a given topic. Therefore, a student group profile should be represented by at least two most likely responses in order to be useful. That is why it is important to define the characteristics of each student group and the

number of student groups as criteria for assessing the usefulness of a set of student group profiles for effective diagnostic feedback.

Once the criteria for assessing the usefulness of a set of student group profiles was defined, the implemented SDNN integrated with a graphical user interface was applied to the two training data sets prepared in the previous section. Using the defined criteria and the insight gained into the relationship among the learning parameters, two set of student group profiles were developed corresponding to the two assessment tasks described in the previous section. The two sets of student group profiles are shown in Figures 6 and 7. The learning parameters used were 2,000 maximum epochs and 0.1 and 0.2 learning rates and 100 number of s nodes, which is large enough for allocating uncommitted nodes and identify up to 100 student groups.

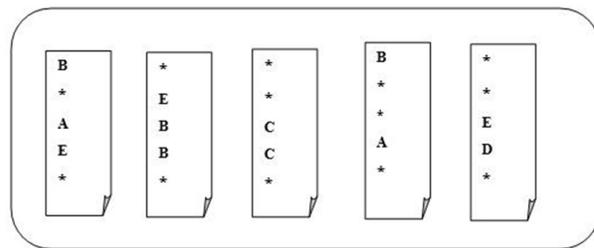


Figure 6. Student group profiles for the first assessment task

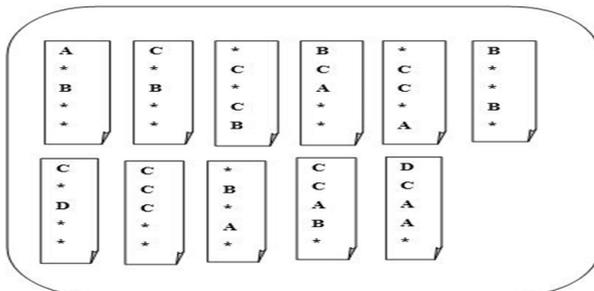


Figure 7. Student group profiles for the second assessment task

7. DIAGNOSTIC FEEDBACK CONSTRUCTION

Feedback is the most essential part of formative assessments as the sole purpose of performing formative assessment is to provide feedback based on students' responses observed from assessments.^[22] Feedback can be defined as an information communicated to a learner that is intended to modify the learner's thinking or behaviour for the purpose of improving learning.^[29] A formative assessment without feedback has no effect on improving student learning experience. On the other hand, a formative assessment with feedback does not

necessarily improve student learning performance.^[30] For example, if answers are included as part of feedback, the effect of feedback on student learning could be negative.^[30,31] The quality of feedback determines the effectiveness of formative assessment.^[30] Several research studies have been conducted to find out what characteristics of feedback actually improve student learning. Most researchers agree that for feedback to be effective, it should be non-evaluative, supportive, timely and specific and should include the comparison of actual performance with some established standard of performance.^[29] Researchers have also reported that the content of effective feedback should contain both verification and elaboration.^[29] Verification indicates whether student's work is correct or not. Elaboration is an information that provides details of how to improve an answer.^[29] The Elaboration aspect of a feedback can be more specific and directive when it addresses a topic, a response or a particular error. On the other hand, it can be more general and facilitative when it provides worked examples or gives gentle guidance.

A diagnostic feedback is a feedback generated based on the analysis of student responses that addresses gaps of understanding and misconceptions.^[20,21] It should identify the gaps of understanding or misconceptions and provide details on how to close the gaps of understanding and correct misconceptions without specifying which questions are answered correctly or not. To support construction of diagnostic feedback, student responses are represented by student group profiles as described in the previous section. Student responses are captured for a particular topic where a set of related multiple choice questions are designed to assess students' conceptual understanding of the topic. Instead of writing for each student response, a diagnostic feedback is constructed per student group profile.

A combination of student responses, which can be extracted from a set of student group profiles, reveals which concepts are already understood and/or which are misunderstood. This information helps tutors in writing the verification part of a diagnostic feedback. Learning outcomes of a particular topic, which specifies what students should be able to do at the end of teaching sessions that covers the topic, define a set of related concepts for the topic and their level of understanding. Tutors can use this information to write the elaboration part of a diagnostic feedback that details how students close gaps of understanding and correct misconceptions. If you consider the first student group profile from the set of student group profiles for the first assessment task shown in Figure 6, most students respond option B for question 1, option A for question 3 and option E for question 4. By looking at this combination of responses, learning outcomes of the assessment task and the multiple choice questions, we can

understand that the students have difficulty in rounding calculated probability values and choosing a right event type, however, they have understood the concept of probability and conditional probability. The second and fifth asterisk (*) indicate that there are no most likely responses for question 2 and 5 respectively. A generic feedback that addresses all incorrect options of both questions can be included. The diagnostic feedback constructed for this student group profile and another diagnostic feedback constructed for the first student group profile of the second assessment task are shown in Tables 1 and 2. The diagnostic feedback for the remaining student group profiles were constructed based on the same procedure described for the first student group profile.

Table 1. Constructed Diagnostic Feedback for the First Student Group of the First Assessment Task

You have understood the concept of probability and conditional probability. However, pay attention to appropriate rounding of calculated probability values and also make sure that the right event is considered. For instance, adult does not mean only male or female, it includes both male and female. If the probability that A occurs given the event B has occurred ($P(A|B)$) is equal to the probability that A occurs ($P(A)$), then A and B are said to be independent or unrelated. For example, if we throw a coin and get a head, this outcome will not affect the outcome of a second throw, as a result these events are independent.

Table 2. Constructed Diagnostic Feedback for the First Student Group of the Second Assessment Task

A constructor is a special method that is called when you create class instances (objects). It is not like any other instance methods since it does not return or change value of fields. Constructors are used to initialise the fields of objects to user input values. They can also set the value of a field which is not specified as parameter to its default value. When you create objects using a constructor, the default values are not supplied as parameters. The header of constructors is similar to method headers, however, the name of a constructor is the same as its class name and no return type is specified explicitly.

8. CONDUCTING ASSESSMENT SESSIONS

A web-based formative assessment tool was designed and implemented to conduct assessment sessions so that students are able to undertake assessment tasks based on multiple choice questions, receive a diagnostic feedback based on their responses instantly, attempt multiple times and record information about the session. The tool integrates five components, which are the implemented snap-drift modal learning neural network, assessment tasks based on multiple choice questions, diagnostic feedback, assessment session manager and a relational database. The tool was implemented using web technologies (XHTML, CSS, Java script and AJAX), Java

technologies (JSP and Servlet), MySQL relational database server and Hibernate for mapping relational tables to Java objects. The implemented snap-drift modal learning neural network first identifies groups which are then represented as student group profiles during training phase and it is also able to assign a student group profile for a new student response. The relational database component stores assessment tasks, constructed diagnostic feedback and information about assessment session such as student responses, attempts, duration between attempts, time and date of sessions. The assessment session manager manages the whole assessment process, which includes displaying an assessment task, capturing student responses, getting the appropriate diagnostic feedback and delivering it to a student, and recording all relevant information about a session.

Two trials were conducted using the developed web-based formative assessment tool during student workshops. During trials, students were not allowed to help each other. A different cohort of students, who were not participated during training phase, were chosen to participate in the trials. All

selected students were registered for Introduction to Data Analysis module for the first trial and Introduction to Programming module for the second trial. Before the trials, they were exposed to the selected topics of the modules. Forty nine students participated during the first trial and twenty five students during the second trial.

9. RESULTS AND DISCUSSION

To assess whether each student response is assigned to an appropriate student group profile, all student responses and their corresponding student group profile were extracted from the database recorded using the web-based formative assessment tool during the two trials. The extracted student responses are shown in Figures 8 and 9 from the first and second trials respectively. Fifteen unique student responses were assigned to student group profile one from the first trial. By comparing the student group profile of student group one with the assigned student responses, six responses matched perfectly with the group profile pattern. For the remaining responses, there are only two matches out of the three for seven responses and one match for two responses.

| Group Number | Group Profile | Response | Group Number | Group Profile | Response | Group Number | Group Profile | Response | |
|--------------|-----------------------|----------|--------------|-----------------------|----------|--------------|-----------------------|----------------------------|-------|
| 1 | B * A E * | BEAEA | 2 | * E B B * | BEBBC | 5 | * * E D * | BEEDA | |
| | | BEADA | | | BEBBB | | | BBEDA | |
| | | CBAEB | | | BEBBA | | | BEEDD | |
| | | BAAEB | | | AEBBD | | | BECD | |
| | | BEAED | | | BEBBD | | | AEEDC | |
| | | BEBEC | | | AEBBB | | | AEADA | |
| | | BEAEB | | | BBBBB | | | BACDC | |
| | | CBCEA | | | BDBBA | | | BEEDC | |
| | | AEAE | | | AEBBB | | | BEEDA | |
| | | BBAED | | | AEBBA | | | BEEDB | |
| | | BEBEB | | | BEEDA | | | BAEDD | |
| | | BEAEC | | | AEBBC | | | DCBDA | |
| | | BEBEA | | | | | | | |
| | | CADEB | | | 4 | | | B * * * A * | BADAB |
| | | BEADB | | | | | | | BEDAD |
| | BCBAC | | | | | | | | |
| | BEDAA | | | | | | | | |
| | BAAAA | | | | | | | | |
| | BADAC | | | | | | | | |
| | BADAB | | | | | | | | |
| | BADAA | | | | | | | | |
| | BEDAD | | | | | | | | |
| | BADAA | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| 3 | * * C C * | CCECC | | | | | | | |
| | | BCECD | | | | | | | |
| | | ABCCA | | | | | | | |
| | | ACCCC | | | | | | | |
| | | | | | | | | | |

Figure 8. Unique student responses extracted and sorted according to their corresponding student group profile for the first trial

The remaining student group profiles were compared from both trials with their respective assigned student responses and the student responses that are partially matched were examined if they could be assigned to any other existing student group profiles. The result showed that none of them could be assigned to any other student group profiles. This implies that the trained snap-drift neural network has managed to assign those partially matched responses to the closest student group as much as possible. The percentage of perfectly matched student responses could be improved by increasing the number of student group profiles and selecting student

group profiles with no more than three most likely responses.

To assess the effectiveness of the diagnostic feedback in improving learning performance of students, data extracted from the database recorded using the web-based formative assessment were analysed. The data set includes the sequence of attempts per each student, the assigned student group profile and their corresponding diagnostic feedback for each attempt and duration between consecutive attempts. A state transition diagram was used to visualise the interaction among the student responses, student group profiles and diagnostic feedback. Figures 10 and 11 show the possible

transition of students for the first trial and second trial respectively. The different student group profiles are identified as states. The state of a student is determined by his/her current responses and a diagnostic feedback is considered as an event that can trigger transition from one student group

profile to another. A student group profile is characterised by a group number and average score. The starting state of a student is determined by the first response and can be at the final state if he/she answers all questions correctly.

| Group Number | Group Profile | Response | Group Number | Group Profile | Response | Group Number | Group Profile | Response |
|--------------|---------------|----------|--------------|---------------|--------------|--------------|---------------|----------|
| 1 | A | ABBAC | 2 | C | CABDA | 3 | * | ACBCB |
| | * | ABBCB | | * | BBBDA | | C | DCBBB |
| | B | ABBBA | | * | CCBCA | | * | BCCCB |
| | * | AABCA | | * | | | C | DDDDB |
| 4 | B | BCAAD | 5 | * | BCCCA | 6 | B | BADBA |
| | C | BCABB | | C | ACCCA | | * | BBBBC |
| | A | BCABD | | C | | | * | BCDBB |
| | * | BCABA | | * | | | B | BCCBB |
| | * | ACACC | 7 | A | | 8 | * | BCBBB |
| | | BCADA | | C | No responses | | C | BCDBC |
| | | BCACA | | * | | | C | |
| | | BCABC | | D | | | C | BCCAC |
| | | BCAAA | | * | | | * | |
| | | BCBCD | | D | | | C | |
| | | BCDCC | | * | | | * | |
| | | BAABA | | * | | | * | |
| | | BCADD | 9 | * | BBBAB | 10 | C | CCADA |
| | | BCAAB | | B | BBAAA | | C | CCACB |
| | | BBACB | | * | CBCAB | | A | CCABB |
| | | BCACB | | A | CBAAB | | B | CCABA |
| | | BCDDA | | * | CBDBA | | * | CAADA |
| | | BCCDD | | | CBCAA | | | CAABB |
| | | BCACD | | | | | | CAAAA |
| | | | | | | | | |
| 11 | D | ACAAB | | | | | | |
| | C | | | | | | | |
| | A | | | | | | | |
| | * | | | | | | | |

Figure 9. Unique student responses extracted and sorted according to their corresponding student group profile for the second trial

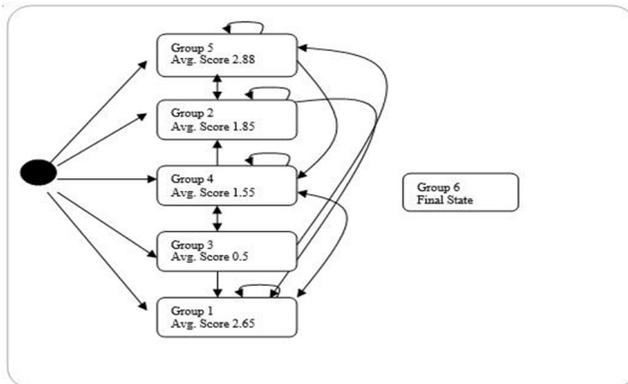


Figure 10. Visualising the possible transition of students from one student group profile to another using transition state diagram for the first trial

From the above state transition diagram shown in Figure 10, we can understand that all student group profiles can be a starting point and students were changing states within the five intermediate states. However, there is no link between the intermediate states and the final state and there is no clear pattern that shows a learning path to reach the final state.

Possible reasons for this might be difficulty of the questions or little effort in reading diagnostic feedback as it was revealed from the information about the durations between consecutive attempts.

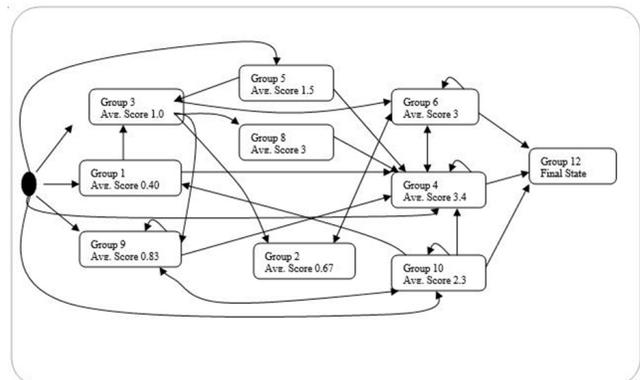


Figure 11. Visualising the possible transition of students from one student group profile to another using transition state diagram for the second trial

There are different starting points in the above state transition diagram shown in Figure 11, which demonstrates that

students were at different group states with different gaps of understanding and misconceptions when they start the assessment session. During the assessment session, some students managed to reach the final state and they followed a pattern to reach the final state. The state transition diagram revealed that students have to reach group states 6, 4, and 10 before they move to the final state. Most students start from low performance (low average score) group states and move toward the high performance group states and finally reach the final state. This indicates that the diagnostic feedback has a positive impact on improving the learning performance of students.

10. CONCLUSION AND FUTURE WORK

Based on related previous research studies, we developed an interactive software tool that implements snap-drift modal learning neural network integrated with visualisation feature of Matlab. The tool is capable of analysing the dynamics of learning behaviour of snap-drift learning algorithm and the effect of learning parameters on the identified student groups.

The proposed method details an algorithm for profiling outputs of snap-drift modal learning neural networks, a criteria for assessing usefulness of student group profiles in revealing gaps of understanding and misconceptions of a particular topic, and a guideline for supporting tutors in writing diagnostic feedback based on profiled student groups. The method was applied to two real assessment tasks designed

for two topics selected from Introduction to Data Analysis and Introduction to Programming modules in order to gather student responses and identify two sets of student group profiles. Two trials were also conducted using a developed web-based formative assessment tool during workshops to evaluate the effectiveness of the proposed method. Forty nine students participated during the first trial and twenty five students during the second trial. Analysis of gathered student responses showed that all of them were assigned to their appropriate student group profiles and the percentage of perfectly matched student responses could be improved by increasing the number of student group profiles and selecting student group profiles with no more than three most likely responses. The analysis of gathered student responses also showed that the diagnostic feedback constructed based on the identified student group profiles has a positive impact on improving the learning performance of students.

Future work will concentrate on developing an intelligent web-based formative assessment for facilitating conceptual understanding of topics based on the proposed method and developed software tools. In addition to this, we will extend the application of snap-drift modal learning neural network to model student responses gathered from programming exercises.

ACKNOWLEDGEMENTS

The authors would like to thank London Metropolitan University for funding this research project.

REFERENCES

- [1] Koutsojannis C, Hatzilgeroudis A. Web based intelligent tutoring system teaching nursing students fundamental aspects of biomedical technology. The 23rd annual EMBS international conference. Istanbul; 2001.
- [2] Nouh Y, Karthikeyani P, Nadarajan R. Intelligent tutoring system-bayesian student model. 2006 1st International Conference on Digital Information Management; 2007. p.257-62.
- [3] Gilmore D, Self J. Artificial Intelligence and Human Learning: Intelligent Computer-Aided Learning. London, UK: Chapman and Hall; 1988.
- [4] Homsy M, Lutfi R, Maria CR, *et al.* Student modeling using nn-hmm for efl course. In: 2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications, Damascus, Syria; 2008. p.1-6.
- [5] Liu H, Shi H, Shang Y. Student modeling with timed assessment information. ITRE 2004 2nd International Conference Information Technology: Research and Education, London, England, UK; 2004. p.121-5.
- [6] Yusob B, Shamsuddin SMH, Ahmad NB. Developing student model using kohonen network in adaptive hypermedia learning system. 2009 Ninth International Conference on Intelligent Systems Design and Applications (ISDA), Pisa, Italy; 2009. p.938-43.
- [7] Stathacopoulou R, Magoulas GD, Grigoriadou M. Neural network-based fuzzy modeling of the student in intelligent tutoring systems. IJCNN'99 International Joint Conference on Neural Networks Intelligent Systems Design and Applications (ISDA), Washington, DC, USA; 1999. p.3517-21.
- [8] Walker R, Voce J, Ahmed J. 2012 survey of technology enhanced learning for higher education in the uk. Ucsa, Tech. Rep.; 2012.
- [9] Sandberk J, Andriessen J. Where is AI and how about education. Artificial Intelligence in Education. 1997: 545-52.
- [10] Wang T. Web-based quiz-game-like formative assessment: Development and evaluation. Computers and Educations. 2008; 51: 1247-63. <http://dx.doi.org/10.1016/j.compedu.2007.11.011>
- [11] Hot potatoes web based assessment tool: accessed on November 30, 2012. Available from: <https://hotpot.uvic.ca/index.php>
- [12] Question mark perception web based assessment tool: accessed on October 12, 2012. Available from: <https://www.questionmark.com/us/pages/default.aspx>
- [13] Sukkarieh JZ, Blackmore J. C-rater: Automatic scoring for short constructed responses. Proceedings of the Twenty-Second International FLAIRS Conference; 2009.

- [14] Sukkarieh JZ, Pulman SG. Information extraction and machine learning: Auto-marking short free text responses to science questions. *Artificial Intelligence in Education*. 2005; 125: 629-37.
- [15] Jordan S, Mitchell T. e-Assessment for learning? the potential of short-answer free-text questions with tailored feedback. *British Journal of Educational Technology*. 2009; 40(2): 371-85. <http://dx.doi.org/10.1111/j.1467-8535.2008.00928.x>
- [16] Murphy C, Kaiser G, Loveland K, *et al.* Retina: Helping students and instructors based on observed programming activities. SIGCSE'09, Tennessee, USA; 2009.
- [17] Blayne P, Freeman M. Individualised interactive formative assessments to promote independent learning. *Journal of Accounting Education*. 2008; 26: 155-65. <http://dx.doi.org/10.1016/j.jaccedu.2008.01.001>
- [18] Fu X, Peltsverger B, Liu J. Apogee-automated project grading and instant feedback system for web based computing. 39th SIGCSE technical symposium on Computer Science Education. Portland, USA: ACM; 2008.
- [19] Truong N, Bancroft P, Roa P. Learning to program through the web. 10th annual SIGCSE conference on Innovation and technology in Computer Science Education. Monte de Caparica, Portugal: ACM; 2005. p.9-13.
- [20] Palmer-Brown D, Lee SW, Jayne C, *et al.* Modal learning neural networks. *WSEAS Transactions on Computers*. 2009; 8(2): 222-36.
- [21] Aleman J, Palmer-Brown D, Jayne C. Effects of response-driven feedback in computer science learning. *Education*. 2011; 54(3): 501-8.
- [22] Bell B, Cowie B. The characteristics of formative assessment in science education. *Science Education*. 2001; 85: 536-53. <http://dx.doi.org/10.1002/sce.1022>
- [23] Kang M, Palmer-Brown D. A modal learning adaptive function neural network applied to hand-written digit recognition. *Information Sciences*. 2008; 178(20): 3802-12. <http://dx.doi.org/10.1016/j.ins.2008.05.011>
- [24] Lee SW, Palmer-Brown D. Phonetic feature discovery in speech using snap-drift. *International Conference on Artificial Neural Networks*, ser. LNCS; 2006. p.952-62.
- [25] Nicol D. E-assessment by design: using multiple-choice tests to good effect. *Journal of Further and Higher Education*. 2007; 31(1): 53-64. <http://dx.doi.org/10.1080/03098770601167922>
- [26] Carneson J, Delpierre G, Masters K. Designing and managing multiple choice questions. *Tech. Rep.*; 2002.
- [27] Eckerdal A, Thune M. Novice java programmers conceptions of object and class, and variation theory. 2005 10th Annual Conference on Innovation and Technology in Computer Science Education, Monte de Caparica, Portugal; 2005.
- [28] Frias-Martinez E, Magoulas G, Chen S, *et al.* Modelling human behaviour in user-adaptive systems: Recent advances using soft computing techniques. *Expert Systems with Applications*. 2005; 29: 320-9. <http://dx.doi.org/10.1016/j.eswa.2005.04.005>
- [29] Shute VJ. Focus on formative feedback. *Review of Educational Research*. 2008; 78: 153-89. <http://dx.doi.org/10.3102/0034654307313795>
- [30] Black P, William D. Assessment and classroom learning. *Assessment in Education: Principles, Policy and Practice*. 1998; 5(1): 7-68. <http://dx.doi.org/10.1080/0969595980050102>
- [31] Bruno I, Santos L. Written comments as a form of feedback. *Studies in Educational Evaluation*. 2010; 36: 111-20. <http://dx.doi.org/10.1016/j.stueduc.2010.12.001>