CLARISSE: A Machine Learning Tool to Initialize Student Models

Esma Aïmeur[1], Gilles Brassard[1], Hugo Dufort[2] and Sébastien Gambs[1]

[1] Université de Montréal, Département IRO
C.P. 6128, Succursale Centre-Ville, Montréal (Québec), H3C 3J7 CANADA
{aimeur,brassard,gambsseb}@iro.umontreal.ca
[2] Netvention Inc., 33 Prince Street, Suite 307
Montreal (Québec), H3C 2M7 CANADA
hdufort@netvention.com

Abstract. The initialization of the student model in an intelligent tutoring system is a crucial issue. It is not realistic to assume that each new student has the same prior knowledge concerning the topic being taught, be it nothing or some "standard" prior knowledge. We introduce CLARISSE, which is a novel categorization method. We illustrate this tool with the identification of categories among students for QUANTI, an intelligent tutoring system for the teaching of quantum information processing. In order to classify a new learner, CLARISSE generates an adaptive pre-test that can identify with high accuracy the learner's category after very few questions.

1 Introduction

In Intelligent Tutoring Systems (ITSs), the *student model* assesses the current state of a student's knowledge and makes inferences about the gaps in his skills. Students want to be active and challenged to reason about the material they are taught. They also need sophisticated feedback, customized curriculum, help and adapted guidance.

To do so, we need to categorize student profiles in order to bring together people who share similar prior knowledge. This makes it possible to focus more quickly on the needs of students. This categorization is important for several reasons. The tutor will be able to select the topics to be taught in a more appropriate manner. Moreover, if a student wishes to exchange his ideas on a given topic with fellow students, it is much easier if they share prior knowledge and common interests.

The initialization of the student model is one of the most important problems that faces ITSs. This initialization could be long-term or short-term [9]. The quality of tutoring depends highly on the relevance of the information acquired during the initialization process. Pre-tests are often used to initialize the student model and it is very convenient when these pre-tests are built per categories.

Quantum mechanics explains the behaviour of elementary particles. Quantum information is very different from its everyday classical counterpart: it cannot be measured reliably and it is disturbed by observation, but it can exist in a superposition of classical states. *Quantum Information Processing* (QIP) is the new and exciting field that studies the implication of quantum mechanics for information processing purposes [8]. This is the realm of futuristic concepts such as quantum cryptography, quantum computing, quantum teleportation and the computation of distributed tasks with vastly reduced communication cost. Some of these ideas are still theoretical, but others have been implemented in the laboratory. In the past few years, QIP has grown tremendously in worldwide interest and activity, especially since Peter Shor's momentous discovery that quantum computers—if only they could be built—would defeat most cryptographic schemes currently in use over the Internet to protect the transmission of sensitive information such as credit card numbers [18], leaving unconditionally secure quantum cryptography as a leading alternative [5].

In this paper, we introduce CLARISSE: a novel tool for initializing the student model. We illustrate the working of CLARISSE in the context of QUANTI [1,2], which is an ITS currently under development for the teaching of quantum information processing. The automatic teaching of QIP is important because this revolutionary new field is still cruelly lacking experts despite all the attention it attracts. More to the point of this paper, however, our main interest in QIP stems from the challenge of categorizing student profiles because of its inherent multidisciplinarity that draws on computer science, mathematics, physics and chemistry.

2 The Curriculum and the Student Model

2.1 The Curriculum

The curriculum is based upon the knowledge base of our Intelligent Tutoring System QUANTI (currently under development), for which a novel web-based elicitation algorithm was developed to help experts during the knowledge acquisition phase [1,2]. The knowledge representation used is a form of semantic network, a graph where nodes (called *entities*) are pieces of knowledge and edges represent *relations* between these nodes.

The network of *concepts* forms the highest level of the knowledge base. Each concept in turn can be broken up into a semantic network. In this network, the concept itself plays the role of *root* in the data structure. The network is made of three different kinds of nodes: components, characteristics and examples. A *component* represents one of the pieces of knowledge that forms the concept. A *characteristic* is generally linked to either a component, in which case it expresses one of its features, or to another characteristic. An *example* serves to illustrate a component.

2.2 The Student Model

The student model is composed of three sub-models: the cognitive model, the affective model and the inferential model.

The *cognitive model* is in charge of representing the learner's knowledge of the domain—what he knows and what he does not know, and to which extent. This part is implemented using an *overlay* model that derives its structure directly from the structure of the curriculum. As the curriculum, the overlay model is made of semantic networks, each being the reflection of their counterparts in the curriculum. Recall that each node is a piece of knowledge. Our goal is to know the level of understanding of the learner. A percentage is associated with each node that represents the level of understanding: 0% means that the user knows nothing (or that we have assumed that he knows nothing) about this particular piece of knowledge, 100% means that he has totally mastered the subject. The main focus of this paper is on how to initialize these values for a new learner.

The *affective model* records the affective profile and the emotional state of the learner. This is an important part in any ITS, but we do not discuss it further because it is peripheral to the issues considered in this paper. The *inferential model* draws inferences about the student from the data available in the cognitive and affective models. In turn, these inferences themselves modify and update these two models.

2.3 Initialization of the Cognitive Model

To initialize the cognitive model, the simplest solution would be to assume that a new learner knows nothing about the domain before starting his first lesson: the level of understanding for each node is set to zero. However, new students are not necessarily unfamiliar with the domain taught by the ITS. Therefore, a better approach is to evaluate the learner with a questionnaire called *pre-test*, which is given before the start of the first session.

For an evaluation to be as accurate as possible, the ideal solution would require to ask at least one question for each node. This is done with an *exhaustive* pre-test. In practice, this method is often too demanding for the student. The number of questions asked of the student during the pre-test could be very high and the student, who is eager to start the course, may feel desperate.

In an *intelligent* pre-test, the questions are focused on the more important nodes. Once these values are measured, the mechanisms of the inferential model are activated in order to propagate these values inside the network. This allows us to reduce the number of questions, but there is a tradeoff between the number of questions in the pre-test and the accuracy of the model. With an intelligent pre-test, the system has to use inferences for the nodes for which it has not asked a question directly to the learner. This reduces the information reliability. The *adaptive* pre-test [3,16], which chooses the next question by taking into account the answers to previous questions, is an example of an intelligent pre-test.

Categorization is another way of avoiding to bury the student under a ton of questions. Each student has unique characteristics and behaviour. However, it is often possible to observe patterns among students and to group students with similar features within *categories*, sometimes called *stereotypes* [14]. Once the categories are discovered, the only task left when a new learner arrives is to be able to determine to which category he belongs. The number of questions required for determining the category of a student is generally much smaller than for an intelligent pre-test. To each category, there corresponds a different initialization of the values of the nodes. This is the approach taken here.

3 Categorization Method

3.1 State of the Art

Categorization is a form of unsupervised learning. We know that we want to find something but we do not know exactly what it should look like. The categories are not known *a priori*: they are revealed by the categorization process. Categorization can be defined as the task of finding structure within data.

Two families of categorization methods exist. The earlier one, which contains the *mathematical and statistical approaches*, does not provide any explanation as to why we end up with these categories or the relations between the items. We concentrate on the second family: the *symbolic and conceptual approaches* [4,7,17]. They try not only to regroup items that are close but also to find how the attributes of the items are similar or different between each other. This makes it possible to provide explanations for the categories thus created. Rules are issued in order to separate items into different groups. Rules can be used recursively for the separation of items. For example, once a rule has been used for splitting a partition of items into two categories, it is possible to choose another rule for the creation of smaller categories within one of these main categories. Among the methods proposed thus far, we note UNIMEM [15], CobWeb [10] and WITT [13].

3.2 CLARISSE

To successfully initialize student models, we use a categorization method. Mathematical clustering by itself is not satisfactory because we want to find categories in an initial set of students, and rules that can be used to categorize new students. For this purpose, we developed a categorization method called CLARISSE, which stands for CLusters And Rules ISSuEd. It works by recursively splitting an initial set of items and building a binary tree that is then used to identify categories.

CLARISSE can be used to process any type of entries; we call each entry an *item*. Items are made of *descriptors*, which are single-valued attributes. The value of each descriptor must be defined, and all items must be comparable in terms of descriptors. Each descriptor has a range of acceptable values, or *domain*. In the usual application of CLARISSE to Intelligent Tutoring Systems, items are *students*, descriptors are *questions* (from a questionnaire), and domains are *question types*.

In CLARISSE, each item can be represented by a point in an N-dimensional space where N is the number of descriptors. If each descriptor has K possible values, then there are $2^K 2^N$ ways of splitting the descriptor space at each step of the method. (This formula must be adapted if the number of possible values is not the same for each descriptor.) CLARISSE reduces the search space to the most promising combinations of descriptors and values.

Great care must be taken when defining domains. In particular, the *semantic distance* between possible values should be defined very precisely. A domain can be defined as a finite range of integer or real values, or as a discrete space of labelled attributes, in which case a matrix of semantic distances must be supplied.

To drive the categorization process, CLARISSE relies on a measure of *category utility* (CU) [12]. This measure tells us how much a category is well defined. According to [6], two factors must be taken into account when calculating the CU: given any item that belongs to some category, the *dissimilarity* indicates how much this item can be differentiated from any item that belongs to another category, whereas the *internal coherence* indicates how much this item is similar to all the other items in the same category. The role of the internal coherence is to prevent the creation of "junkyard" categories that contain orphaned items.

CLARISSE derives the internal coherence of a category from a measure of the entropy in the descriptors. In order to reduce this entropy, items can be *swapped* between categories, but there is a price associated to this action: our measure of internal coherence *and* our measure of dissimilarity might suffer from it. Since it is recursive and it can use backtracking, CLARISSE can try various ways of partitioning clusters, which yield slightly different results that are sorted by category utility.

Here is a sketch of how CLARISSE works.

- Start with an initial set M containing *n* items.
- To execute a mathematical clustering, we need to select two prototypes (or seeds) among the *n* items in M. We choose S_1 and S_2 as the farthest items in the descriptor space. This is one of many possible strategies. This operation takes time in $O(n^2)$.
- Apply mathematical clustering (aggregation) to S_1 and S_2 by use of either the centroid method, the farthest neighbour method, or the nearest neighbour method, to obtain clusters C_1 and C_2. The complexity of these methods ranges from $O(n^2)$ to $O(n^3)$, and affects the properties of clusters in ways that we shall not discuss here. See [13].
- Choose a descriptor D_0 whose values can be partitioned in a way that maximizes the dissimilarity between C_1 and C_2. Once such a descriptor is found, try to find additional descriptors D_i that can be used with D_0 to refine our partition of the descriptor space, without critically reducing the quality of C_1 and C_2. The conjunction $\{D_0 \wedge ... \wedge D_i\}$ of such descriptors is called a *rule*, which we note R.
- If some items in C_1 and C_2 are cut from their parent cluster by the application of R, then try to swap these items between C_1 and C_2. As we said above, there is a price associated with this swapping: it is the loss of internal coherence in the receiving cluster. This price, which we call *item displacement cost*, must be counterbalanced by the gain in overall partition quality. We weigh this price using an internal parameter, which we call *belonging weight*.
- After trying all the rules that look promising, keep the best candidate. Swap items between C_1 and C_2. The candidate rules (that were not used) are kept in a set, and are available in case we choose to backtrack and try another path.
- Recursively apply these steps on C_1 and C_2. If some category contains only one item, or all of its items are indistinguishable, that category is final.

In Fig. 1, two mathematical clusters (represented by dashed ovals) were built around seeds S_0 and S_1. CLARISSE found a rule on *Y*: *all items with* $Y < 5$ *are in category* C_1 *and all items with* $Y \geq 5$ *are in category* C_0. Three *fringe items* (represented by white circles) were swapped in the process. The resulting categories are both coherent and explainable.

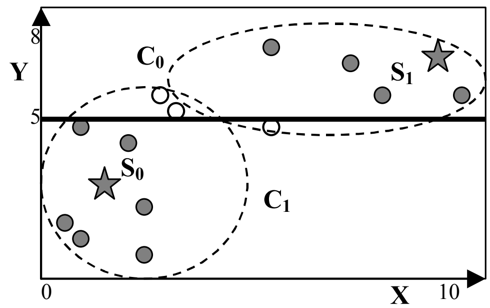

Fig. 1. Example of a partition in the R×R plane

4 The Experiment: Methodology and Analysis of the Results

To identify student categories in quantum information processing, we used a sample of the target public as training set. We built a questionnaire containing 30 multiple-choice questions covering themes ranging from classical logic and matrices to advanced quantum information processing topics. Answers were rated using three values: good (10 points), bad (3 points), very bad (0 points). These ratings correspond to the descriptor space distances used in the clustering algorithm. Each question is used as a descriptor in the clustering process, and thus in our case CLARISSE is working in a 30-dimensional descriptor space.

Students, teachers and researchers (in physics and in computer science) were asked to participate on a voluntary basis through a web-based questionnaire using HTML pages and JavaScript. Over a one-week period, we received 31 answers from Australia, Canada, France, Israel, Japan, the Netherlands and the United States. Results were passed through a validation program, and formatted as an input file to CLARISSE.

Of course, we are aware that this initial experiment can be seen at best as a way to sharpen our tools. Indeed, the complete classification of students in a field as rich as quantum information processing would require a far greater number of questions as well as data from at least as many participants as there are questions. It would be unrealistic to base the student model of an Intelligent Tutoring System entirely on so little data. Nevertheless, the results obtained by CLARISSE were used to validate the approach in a convincing way, pending a more thorough experiment.

Using its built-in heuristics for inducing variations in its parameters, CLARISSE found only one stable cluster tree, which involved seven clusters. The resulting solution leads us to challenge two well-established beliefs, but care must be exercised given that our current experiment was not conducted on a sufficiently large scale.

Myth #1: *Categories must represent the students' technical background.*

No strong correlation was found between a student's technical background and his results. Profiling students by merely asking them about their technical background, the courses they have taken during the previous year, or their current activities, is not precise enough for an ITS to be efficient. Tutoring is about fulfilling the student's pedagogical needs, and these needs can vary greatly among students with the same technical or pedagogical background.

Myth #2: *Categories must represent partitions in the distribution of scores.*
We found only a weak correlation between scores and categories: the weakest students were grouped in two well-defined categories, but the remaining students were so tightly grouped that any *ad hoc* method would fail to detect subgroups. For example, if we examine the questionnaires whose scores fall in the 80%-90% range, we find that even though the standard deviation is small, different people have made different mistakes. Some of these errors are repeated across a significantly large group of questionnaires; by observing this trend, the presence of categories is revealed.

It turns out that raw results, as used in traditional tutoring, are not giving much information when we try to categorize students. Most teachers have time constraints that prevent them from looking for patterns in their pupils' results. Machine learning tools, such as CLARISSE, can greatly enhance the thoroughness of their analysis.

4.1 The Cluster Definition File

Using analysis of the coherence variation through the cluster tree, CLARISSE partitioned it into 7 categories that showed high coherence. Each category is identified by:
- A value of internal coherence. Generally, a high coherence shows a high density of the category's descriptor space.
- Its list of members from the original set.
- An ordered list of rules that oppose this category to all other categories that were found. These rules form a hierarchical rule system, which can be used to quickly classify a new student.
- A list of the most significant descriptors. These descriptors are used to initialize the student model in an ITS, whenever a student falls into this category.

4.2 Classifying New Students

The main purpose of this process is to classify new students who wish to follow a course in quantum information processing. We define inclusion/exclusion rules for each cluster, which helps us build a hierarchical rule system. For example, there could be an *inclusion* rule for some category that would state that any student who gave a very bad answer to some specific question is included in that category, or an *exclusion* rule that excludes any student who gave the good answer to that same question. It turns out that only 6 of the 30 questions in the original questionnaire are used in these rules. These questions were marked as highly significant by CLARISSE, since they were used to discriminate between emerging categories.

Fig. 2 shows this rule system. Categories are ordered from those that contain highest-ranking scores to those that contain lowest-ranking scores. Failing to validate a capability sends you lower in the decision tree. For each category, a description of the typical members is given. To classify a new student in one of the seven categories, it suffices to ask him questions on at most three concepts with the adaptive pre-test that derives from Fig. 2. Once classified, the student receives the category's standard profile (a set of descriptors) and thus we can initialize his student model in our ITS.

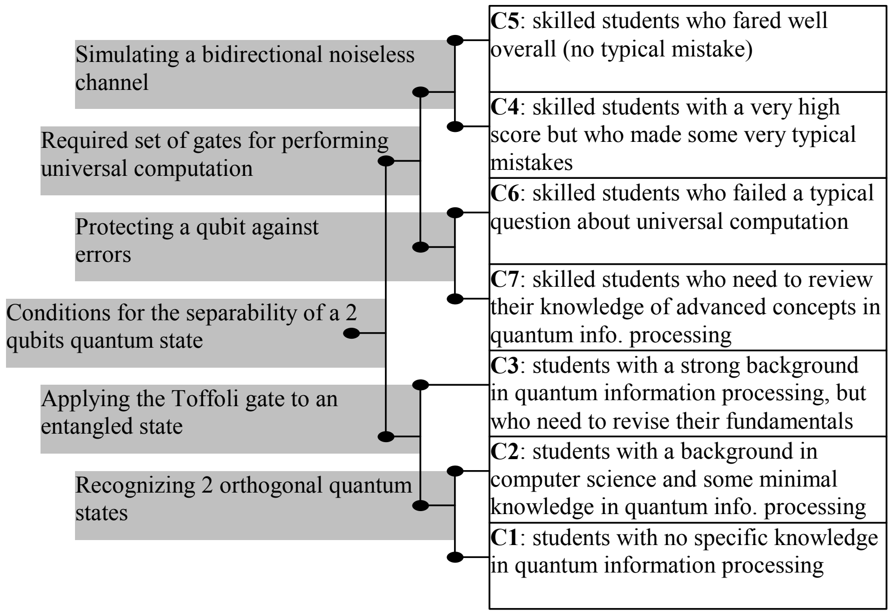

Fig. 2. Decision tree used to classify new students

Because we received four filled questionnaires *after* compiling these results, we decided to validate our method by using the six significant descriptors to classify the new participants. Table 1 shows an evaluation of our results. In two cases, we had a perfect match between the participant's descriptors and the category's descriptors, while in the other two, we had a lower level of success. With a success rate of 61.5%, 26.6% (8/30) of the third participant's nodes in QUANTI would have been initialized incorrectly. This is much better than no initialization at all, and wrong values will eventually be corrected by the inferential agent. The fourth participant has a better fit, with an 88.2% success rate.

Table 1. New students are categorized and their student model is initialized

Student	Category	Number of descriptors in the category definition	Number of descriptors that matched the participant's answers	Success rate
A	C7 - skilled	15	15	100%
B	C5 - highly skilled	16	16	100%
C	C2 - basic knowledge	13	8	61.5%
D	C6 - skilled	17	15	88.2%

Student C in Table 1 somehow falls between categories 1 and 2, and would have been best classified in category 1. When encountering such a case, CLARISSE should be asked to integrate this *hard to classify* student in its training set. Thus it is important to enrich the categories with new students from the population whenever possible, for instance by adding some students on the training set on a regular basis, or by asking a few students to pass the pre-test now and then.

Now that seven categories have clearly been identified, the initial values of the student model have to be computed for each of these categories. Once this phase is completed, the only task left is to use the decision tree in order to recognize to which category each new student belongs. The decision tree plays the role of a small adaptive pre-test whose goal is to classify a student in one of the seven categories. The battery of questions asked of a student is not fixed: the next question asked may depend on the answers to the previous questions.

Each question in the questionnaire is linked to one or several nodes in the cognitive model. This allows us to initialize the student's level of understanding for those nodes. If more than one question point to a node, a weighted average is calculated to set the level of understanding for that node. Once this phase is completed, the inferential model is used to propagate values inside the cognitive model. Once all the initial values for a particular category have been processed, they are stored. When a new learner is detected to belong to that category, his student model is initialized accordingly.

5 Conclusions

Nowadays, a significant amount of research focuses on the enhancement of learning effectiveness of web-based educational systems, which increases the likelihood that distance learners might benefit from this technology-based approach to education.

The main purpose of a student model is to provide the tutor with the information necessary to select a suitable instructional action. The initialization of the student model is one of the most important problems that faces Intelligent Tutoring Systems. It can be complex and difficult, especially when students come from different areas and do not share the same prior knowledge.

In this paper, we introduced CLARISSE, an efficient machine learning tool to categorize student models. The application of CLARISSE to quantum information processing identified seven well-defined categories of students, each having a different set of values for the cognitive model. The process allowed us to challenge two well-established myths. CLARISSE also provided an adaptive pre-test that can classify a student in one of these categories with at most three questions. This is much fewer than would have been required by an exhaustive pre-test, or even compared to the 30 questions in the original questionnaire used in the experimentation.

In earlier work [11], CLARISSE had been tested on various other clustering problems, yielding promising results. We used it to build an identification key for mushrooms, to categorize 120 countries using social development indicators, and to categorize students for various other ITSs such as a spreadsheet tutor and a racquetball tutor.

References

1. Aïmeur, E., Blanchard, E., Brassard, G., Fusade, B. and Gambs, S., “Designing a Multidisciplinary Curriculum for Quantum Information Processing”, *Proceedings of Artificial Intelligence in Education: AI-ED’01*, pp. 524–526, 2001.
2. Aïmeur, E., Blanchard, E., Brassard, G. and Gambs, S., “QUANTI: A Multidisciplinary Knowledge-Based System for Quantum Information Processing”, *Proceedings of International Conference on Computer Aided Learning in Engineering Education: CALIE’01*, pp. 51–57, 2001.
3. Arroyo, I., Conejo, R., Guzmand, E. and Woolf, B.P., “An Adaptive Web-based Component for Cognitive Ability Estimation”, *Proceedings of Artificial Intelligence in Education: AI-ED’01*, pp. 456–466, 2001.
4. Bauer, M., Gmytrasiewicz, P. and Pohl, W., Workshop “Machine Learning for User Modeling”, *Proceedings of the Seventh International Conference on User Modeling*, 1999.
5. Bennett, C.H., Brassard, G. and Ekert, A.K., “Quantum Cryptography”, *Scientific American*, pp. 164–171, October 1992.
6. Biswas, G., Weinberg, J. and Fisher, D., “ITERATE: A Conceptual Clustering Algorithm that Produces Cohesive Clusters”, 1995, available on the Internet at URL: http://cswww.vuse.vanderbilt.edu/~biswas/Papers/kdd/iterate-oilabs.html .
7. Bloedorn, E., Mani, I. and MacMillan, T.R., “Machine Learning of User Profiles: Representational Issues”, *Proceedings of the National Conference on Artificial Intelligence*, pp. 433–438, 1997.
8. Chuang, I.L. and Nielsen, M.A., *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.
9. De Koning, K., and Bredeweg, B., “Exploiting Model-Based Reasoning in Educational Systems”, in *Smart Machines in Education,* K.D. Forbus and P.J. Feltovich (Eds), pp. 299–330, 2001.
10. Fisher, D., “Knowledge Acquisition via Incremental Conceptual Clustering”, *Machine Learning* **2**, pp. 139–172, 1987.
11. Dufort, H., *Évaluation et adaptation automatique de cours dans un système tutoriel intelligent*, Masters thesis under the direction of Esma Aïmeur, Université de Montréal, 1999.
12. Gluck, M.A. and Corter, J.E., “Information, Uncertainty, and the Utility of Categories”, *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, Hillsdale: Lawrence Erlbaum Associates, pp. 283–287, 1985.
13. Hanson S.J., “Conceptual Clustering and Categorisation: Bridging the Gap between Induction and Causal Models”, *Machine Learning: An Artificial Intelligence Approach* **3**, 1990.
14. Kay, J., “Stereotypes, Students Models and Scrutability”, *Proceedings of Intelligent Tutoring Systems: ITS 2000*, pp. 19–30, 2000.
15. Lebowitz, M., “Experiments with Incremental Concept Formation: UNIMEM”, *Machine Learning* **2**, pp. 103–138, 1987.
16. Millàn, E., Pérez-de-la-Cruz, J.L. and Svàzer, E., “Adaptive Bayesian Networks for Multilevel Student Modelling”, *Proceedings of Intelligent Tutoring Systems: ITS 2000,* pp. 534–543, 2000.
17. Raskutti, B. and Beitz, A., “Acquiring User Preferences for Information Filtering in Interactive Multi-Media Services”, *Proceedings of PRICAI*, pp. 47–58, 1996.
18. Shor, P.W., “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”, *SIAM Journal on Computing* **26**, pp. 1484–1509, 1997.