Educational Data Mining: a Case Study

Agathe MERCERON[*] and Kalina YACEF[+]
[*]*ESILV - Pôle Universitaire Léonard de Vinci, France*
[+]*School of Information Technologies - University of Sydney, Australia,*
Agathe.Merceron@devinci.fr, kalina@it.usyd.edu.au

Abstract. In this paper, we show how using data mining algorithms can help discovering pedagogically relevant knowledge contained in databases obtained from Web-based educational systems. These findings can be used both to help teachers with managing their class, understand their students' learning and reflect on their teaching and to support learner reflection and provide proactive feedback to learners.

1 Introduction

Web-based educational systems collect large amounts of student data, from web logs to much more semantically rich data contained in student models. Whilst a large focus of AIED research is to provide adaptation to a learner using the data stored in his/her student model, we explore ways to mining data in a more collective way: just as a human teacher can adapt to an individual student, the same teacher can also learn more about how students learn, reflect and improve his/her practice by studying a group of students.

The field of Data Mining is concerned with finding new patterns in large amounts of data. Widely used in Business, it has scarce applications to Education. Of course, Data Mining can be applied to the business of education, for example to find out which alumni are likely to make larger donations. Here we are interested in mining student models in a pedagogical perspective. The goal of our project is to define how to make data possible to mine, to identify which data mining techniques are useful and understand how to discover and present patterns that are pedagogically interesting both for learners and teachers.

The process of tracking and mining such student data in order to enhance teaching and learning is relatively recent but there are already a number of studies trying to do so and researchers are starting to merge their ideas [1]. The usefulness of mining such data is promising but still needs to be proven and stereotypical analysis to be streamlined. Some researchers already try and set up some guidelines for ensuring that ITS data can be usefully minable [2] out of their experience of mining data in the project LISTEN [3].

Some directions start to emerge. Simple statistics, queries or visualisation algorithms are useful to give to teachers/tutors an overall view of how a class is doing. For example, the authors in [4] use pedagogical scenarios to control interactive learning objects. Records are used to build charts that show exactly where each student is in the learning sequence, thus offering to the tutor distant monitoring. Similarly in [5], students' answers to exercises are recorded. Simple queries allow to show charts to teachers/tutors of all students with the exercises they have attempted, they have successfully solved, making tutors aware of how students progress through the course. More sophisticated information visualisation techniques are used in [6] to externalise student data and generate pictorial representations for course instructors to explore. Using features extracted from log data and marks obtained in the final exam, some researchers use classification techniques to predict student performance fairly accurately [7]. These allow tutors to identify students at risk and provide advice ahead of the final exam. When student mistakes are recorded, association rules algorithms can be used to find mistakes often associated together [8]. Combined with a genetic algorithm, concepts mastered together can be identified using student scores[9].

The teacher may use these findings to reflect on his/her teaching and re-design the course material.

The purpose of this paper is to synthesize and share our various experiences of using Data Mining for Education, especially to support reflection on teaching and learning, and to contribute to the emergence of stereotypical directions. Section 2 briefly presents various algorithms that we used, section 3 describes our data, section 4 describes some patterns found and section 5 illustrates how this data is used to help teachers and learners. Then we conclude the paper.

2 Algorithms and Tools

Data mining encompasses different algorithms that are diverse in their methods and aims [10]. It also comprises data exploration and visualisation to present results in a convenient way to users. We present here some algorithms and tools that we have used. A data element will be called an individual. It is characterised by a set of variables. In our context, most of the time an individual is a learner and variables can be exercises attempted by the learner, marks obtained, scores, mistakes made, time spent, number of successfully completed exercises and so on. New variables may be calculated and used in algorithms, such as the average number of mistakes made per attempted exercise.

Tools: We used a range of tools. Initially we worked with Excel and Access to perform simple SQL queries and visualisation. Then we used Clementine[11] for clustering and our own data mining platform for teachers, Tada-Ed [12], for clustering, classification and association rule (Clementine is very versatile and powerful but Tada-Ed has pre-processing facilities and visualisation of results more tailored to our needs). We used SODAS [13] to perform symbolic data analysis.

Data exploration and visualisation: Raw data and algorithm results can be visualised through tables and graphics such as graphs and histograms as well as through more specific techniques such as symbolic data analysis (which consists in creating groups by gathering individuals along one attribute as we will see in section 4.1). The aim is to display data along certain attributes and make extreme points, trends and clusters obvious to human eye.

Clustering algorithms aim at finding homogeneous groups in data. We used k-means clustering and its combination with hierarchic clustering [10]. Both methods rest on a distance concept between individuals. We used Euclidian distance.

Classification is used to predict values for some variable. For example, given all the work done by a student, one may want to predict whether the student will perform well in the final exam. We used C4.5 decision tree from TADA-Ed which relies on the concept of entropy. The tree can be represented by a set of rules such as: *if* $x=v_1$ *and* $y> v_2$ *then* $t= v_3$. Thus, depending on the values an individual takes for, say the variables x and y, one can predict its value for t. The tree is built taking a representative population and is used to predict values for new individuals.

Association rules find relations between items. Rules have the following form: *X -> Y, support* 40%, *confidence* 66%, which could mean '*if students get X incorrectly, then they get also Y incorrectly*', with a support of 40% and a confidence of 66%. Support is the frequency in the population of individuals that contains both *X* and *Y*. Confidence is the percentage of the instances that contains *Y* amongst those which contain *X*. We implemented a variant of the standard Apriori algorithm [14] in TADA-Ed that takes temporality into account. Taking temporality into account produces a rule *X->Y* only if exercise *X* occurred before *Y*.

3 A case study: Logic-ITA student data

We have performed a number of queries on datasets collected by the Logic-ITA to assist teaching and learning. The Logic-ITA is a web-based tutoring tool used at Sydney

University since 2001, in a course taught by the second author. Its purpose is to help students practice logic formal proofs and to inform the teacher of the class progress [15].

3.1 Context of use

Over the four years, around 860 students attended the course and used the tool, in which an exercise consists of a set of formulas (called premises) and another formula (called the conclusion). The aim is to prove that the conclusion can validly be derived from the premises. For this, the student has to construct new formulas, step by step, using logic rules and formulas previously established in the proof, until the conclusion is derived. There is no unique solution and any valid path is acceptable. Steps are checked on the fly and, if incorrect, an error message and possibly a tip are displayed. Students used the tool at their own discretion. A consequence is that there is neither a fixed number nor a fixed set of exercises done by all students.

3.2 Data stored

The tool's teacher module collates all the student models into a database that the teacher can query and mine. Two often queried tables of the database are the tables *mistake* and *correct_step*. The most common variables are shown in ***Table 1***.

Table 1. Common variables in tables mistake and correct_ step

login	the student's login id
qid	the question id
mistake	the mistake made
rule	the logic rule involved/used

line	the line number in the proof
startdate	date exercise was started
finishdate	date exercise was finished (or 0 if unfinished)

4 Data Mining performed

Each year of data is stored in a separate database. In order to perform any clustering, classification or association rule query, the first action to take is to prepare the data for mining. In particular, we need to specify two aspects: (1) what element we want to cluster or classify: students, exercises, mistakes? (2) Which attributes and distance do we want to retain to compare these elements? An example could be to cluster students, using the number of mistakes they made and the number of correct steps they entered. Tada-ed provides a pre-processing facility which allows to make the data minable. For instance, the database contains lists of mistakes. If we want to group that information so that we have one vector per student, we need to choose how the mistakes should be aggregated. For instance we may want to consider the total number of mistakes, or the total number of mistakes per type of mistake, or a flag for each type of mistake, and so on.

4.1 Data exploration

Simple SQL queries and histograms can really allow the teacher get a first overview of the class[8, 15]: what were the most common mistakes, the logic rules causing the most problems? What was the average number of exercises per student? Are there any student not finishing any exercise? The list goes on.

To understand better how students use the tool, how they practice and how they come to master both the tool and logical proofs, we also analysed data, focussing on the number of attempted exercises per student. In SODAS, the population is partitioned into sets called symbolic objects. Our symbolic objects were defined by the number of attempted exercises and were characterized by the values taken for these newly calculated variables: the number of successfully completed exercises, the average number of correct steps per attempted exercise, the average number of mistakes per attempted exercise. We obtained a number of tables to compare all these objects. An example is given in Table 2, which compares objects according to the number of successfully completed exercises.

Table 2. Distribution of students according to the number of attempted exercises (row) and the number of completed exercises (column) for year 2002.

Finish/Attempt	0	1	2	3	4	5	6	7	8	9	10	11	12	14	15	16	19	20	21	26
1	46	54																		
2	13	23	65																	
3	6	11	39	44																
4-6	4	8	27	19	29	10	2													
7-10	3		6	18	36	12	18	3	3											
11-15			16	16	16	21	5	5			11		5	5						
16 +			17												17		17	33		17

For example, the second line says that, among the students who have attempted 2 exercises, 13% could not complete any of them, 23% could complete one and 65% could complete both. And similarly for the other lines.

Using all the tables, we could confirm that the more students practice, the more successful they become at doing formal proofs[16]. Interestingly though, there seems to be a number of exercises attempted bove which a large proportion of students finish most exercises. For 2002, as little as two attempted exercises seem to put them on the safe side since 65% of the students who attempted 2 exercises were able to finish them both.

4.2 Association rules

We used association rules to find mistakes often occurring together while solving exercises. The purpose of looking for these associations is for the teacher to ponder and, may be, to review the course material or emphasize subtleties while explaining concepts to students. Thus, it makes sense to have a support that is not too low. The strongest rules for 2004 are shown in Table 3. The first association rule says that if students make mistake *Rule can be applied, but deduction incorrect* while solving an exercise, then they also made the mistake *Wrong number of line references given* while solving the same exercise. Findings were quite similar across the years (2001 to 2003).

Table 3. Association rules for Year 2004.

M11 ==> M12 [sup: 77%, conf: 89%] M12 ==> M11 [sup: 77%, conf: 87%] M11 ==> M10 [sup: 74%, conf: 86%] M10 ==> M12 [sup: 78%, conf: 93%] M12 ==> M10 [sup: 78%, conf: 89%] M10 ==> M12 [sup: 74%, conf: 88%]	M10: Premise set incorrect M11: Rule can be applied, but deduction incorrect M12: Wrong number of line reference given

4.3 Clustering and visualisation

We applied clustering to try and characterize students with difficulties. We looked in particular at those who attempted an exercise without completing it successfully. To do so, we performed clustering using this subpopulation, both using (i) k-means in TADA-Ed, and (ii) a combination of k-means and hierarchical clustering of Clementine. Because there is neither a fixed number nor a fixed set of exercises to compare students, determining a distance between individuals was not obvious. We calculated and used a new variable: the total number of mistakes made per student in an exercise. As a result, students with similar frequency of mistakes were put in the same group. Histograms showing the different clusters revealed interesting patterns. Consider the histogram shown in Figure 1 obtained with TADA-Ed. There are three clusters: 0 (red, on the left), 1 (green, in the middle) and 4 (purple, on the right). From other windows (not shown) we know that students in cluster 0 made many mistakes per exercise not finished, students in cluster 1 made few mistakes and students in cluster 4 made an intermediate number of mistakes. Students making many mistakes use also many different logic rules while solving exercises, this is shown with the vertical, almost solid lines. On the other hand, another histogram (Figure 2) which displays exercises against students, tells us that students from group 0 or 4 have not attempted more exercises than students from group 1, who make few mistakes. This suggests that these

students try out the logic rules from the pop-up menu of the tool one after the other while solving exercises, till they find one that works.

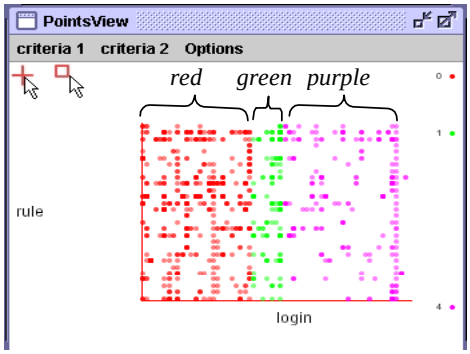

Figure 1. Histogram showing, for each cluster of students, the rules incorrectly used per student

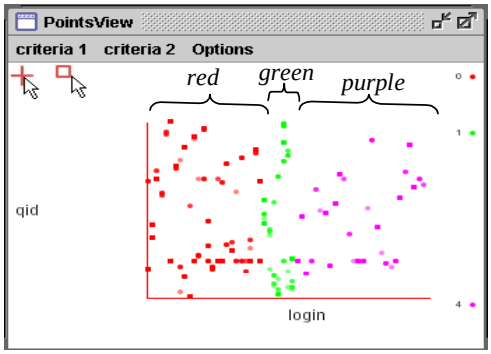

Figure 2. Histogram showing, for each cluster of students, the exercise attempted per student

Note: Since the article is printed in black and white, we superimposed information about where the colors are located.

4.4 Classification

We built decision trees to try and predict exam marks (for the question related to formal proofs). The Decision Tree algorithm produces a tree-like representation of the model it produces. From the tree it is then easy to generate rules in the form `IF condition THEN outcome`. Using as a training set the previous year of student data (mistakes, number of exercises, difficulty of the exercises, number of concepts used in one exercise, level reached) as well as the final mark obtained in the logic question), we can build and use a decision tree that predicts the exam mark according to the attributes so that they can be used the following year to predict the mark that a student is likely to obtain.

Table 4. Some results of decision tree processing. Accuracy of mark prediction using simple rounding of the mark (on 84 students).

Attributes and type of pre-processing	**Accuracy of mark**	**Accuracy of pass/fail**	**Diff. Avg (sd) real/predicted**	**Rel. error**
Number of distinct rules in each exercise* Number of exercises per performance type^	51.9%	73.4%	-0.2 (1.7)	11%
Number of distinct rules* Sum of lines entered correctly in each exercise	46.8%	87.3%	-0.5 (1.9)	18%
Number of exercises per nb of rules (interval)* Different performance achieved ^	45.6%	86.1%	-0.4 (1.8)	14%
Number of different length of exercises# Different performance achieved ^	43%	88.6%	0.14 (1.5)	8%
Number of exercises per performance type^ Sum of lines entered correctly in each exercise	44.3%	86.1%	-0.3 (1.7)	13%
Number of exercises per performance type^ Sum of rules used correctly (incl. repetition)	44%	86.1%	0.1 (1.9)	10%
Sum of rules used correctly (incl. repetition)	43%	87.3%	-0.22 (1.8)	13%
Sum of lines entered correctly in each exercise	43%	87.3%	-0.22 (1.8)	13%
Mistakes, in any form of pre-processing	<20%			

* in order to avoid overfitting we have grouped number of rules into intervals: [0-5], [6-10], [10+] .
\# for the same reason, the number of steps in exercises was grouped into intervals of 5.
^ Performance types were grouped into 3 types: unfinished, finished with mistakes, finished without mistake.

There are a very large number of possible trees, depending on which attributes we choose to do the prediction and how we use them (ie the type of pre-processing we use). We investigated this on different combinations, using 2003 year as training data (140 students) and 2004 year as test data (84 students). After exam results, the 2004 population did very slightly better than the 2003 one, but not with a statistical difference. For each combination we calculated accuracy at different granularity. Table 4 shows some of the results we obtained: the second column shows the percentage of mark accuracy (a prediction is deemed accurate when the rounded value predicted coincides with the real mark). The third

column shows the percentage of accuracy of pass/fail predictions. The fourth column shows the average difference between the predicted exam value and the real exam value, and the standard deviations (which are the same as the root mean squared prediction error). The last column shows the relative squared error. Marks ranged from 0 to 6.

The most successful predictors seemed to be the number of rules used in an exercise, the number of steps in exercises and whether or not the student finished the exercises. Interestingly, these attributes seemed to be more determining than the mistakes made by the student, regardless of how we pre-process them.

5 Supporting teachers and learners

5.1 Pedagogical information extracted

The information extracted greatly assisted us as teachers to better understand the cohort of learners. Whilst SQL queries and various histograms were used during the course of the teaching semester to focus the following lecture on problem areas, the more complex mining was left for reflection between semesters.

- Symbolic data analysis revealed that if students attempt at least two exercises, they are more likely to do more (probably overcoming the initial barrier of use) and complete their exercises. In subsequent years we required students to do at least 2 exercises as part of their assessment (a very modest fraction of it).
- Mistakes that were associated together indicated to us that the very concept of formal proofs (ie the structure of each element of the proof, as opposed to the use of rules for instance) was a problem. In 2003, that portion of the course was redesigned to take this problem into account and the role of each part of the proof was emphasized. After the end of the semester, mining for mistakes associations was conducted again. Surprisingly, results did not change much (a slight decrease in support and confidence levels in 2003 followed by a slight increase in 2004). However, marks in the final exam continued increasing. This leads us to think that making mistakes, especially while using a training tool, is simply part of the learning process and was supported by the fact that the number of completed exercises per student increased in 2003 and 2004.
- The level of prediction seems to be much better when the prediction is based on exercises (number, length, variety of rules) rather than on mistakes made. This also supports the idea that mistakes are part of the learning process, especially in a practice tool where mistakes are not penalised.
- Using data exploration and results from decision tree, one can infer that if students do successfully 2 to 3 exercises for the topic, then they seem to have grasped the concept of formal proof and are likely to perform well in the exam question related to that topic. This finding is coherent with correlations calculated between marks in the final exam and activity with the Logic Tutor and with the general, human perception of tutors in this course. Therefore, a sensible warning system could look as follows. Report to the lecturer in charge students who have completed successfully less than 3 exercises. For those students, display the histogram of rules used. Be proactive towards these students, distinguishing those who use out the pop-up menu for logic rules from the others.

5.2 ITS with proactive feedback

Data mining findings can also be used to improve the tutoring system. We implemented a function in Tada-Ed allowing the teacher to extract patterns with a view to integrate them in the ITS from which the data was recorded. Presently this functionality is available for Association Rule module. That is, the teacher can extract any association rule. Rules are then saved in an XML file and fed into the pedagogical module of the ITS. Along with the pattern, the teacher can specify an URL that will be added to the feedback window and where the teacher can design his/her own proactive feedback for that particular sequence of

mistakes. The content of the page is up to the teacher. For instance for the pattern of mistakes A, B -> C, the teacher may want to provide explanations about mistakes A and B (which the current student has made) and review underlying concepts of mistake C (which the student has not yet made).

```
- <AssociationRules>
  - <AssociationRule>
      <left>Invalid justification</left>
      <left>Premise set incorrect</left>
      <right>Wrong number of line references given.</right>
      <ruleSupport>47.154472</ruleSupport>
      <ruleConfidence>74.35898</ruleConfidence>
      <ruleLift>0.87106234</ruleLift>
      <url>www.ug.usyd.edu.au/~logic/...</url>
    </AssociationRule>
  </AssociationRules>
```

Figure 3. XML encoded patterns

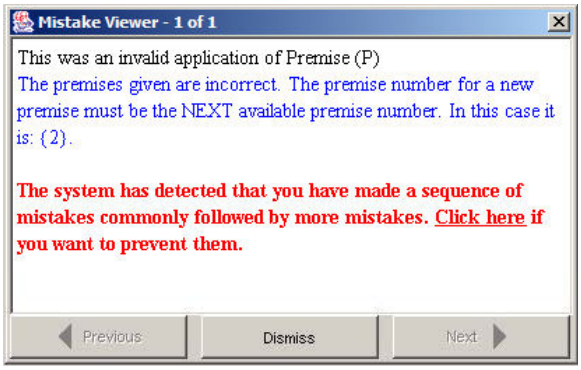

Figure 4. Screen shot of mistake viewer

The structure of the XML file is fairly simple and is shown in ***Figure 3***. For instance, using our logic data, we extracted the rule saying that if a student makes the mistakes "Invalid justification" followed by "Premise set incorrect" then s/he is likely to make the mistake "Wrong number of references lines given" in a later step (presently there is no restriction on the time window). This rule has a support of 47% and a confidence of 74%. The teacher, when saving the pattern, also entered an URL to be prompted to the student.

The pedagogical module of the Logic Tutor then reads the file and adds the rule to its knowledge base. Then, when the student makes these two initial mistakes, s/he will receive, in addition to the relevant feedback on that mistake, an additional message in the same window (in a different color) advising him/her to consult the web page created by the teacher for this particular sequence of mistakes. This is illustrated by ***Figure 4***.

This allows the tutoring system to send proactive messages to learners in order to try and prevent mistakes likely to occur later, based on patterns observed with real students.

5.3 Support for student reflection

Extracting information from a group of learners is also extremely relevant to the learner themselves. The fact that learner reflection promotes learning is widely acknowledged [17]. The issue is how to support it well. A very useful way to reflect on one's learning is to look up what has been learned and what has not yet been learned according to a set of learning goals, as well as the difficulties currently encountered. We are seeking here to help learners to compare their achievements and problems in regards to some important patterns found in the class data. For instance, using a decision tree to predict marks, the student can predict his/her performance according to his/her achievements so far and have the time to rectify if needed. Here more work needs to be done to assess how useful this prediction is for the student.

6 Conclusion

In this paper, we have shown how the discovery of different patterns through different data mining algorithms and visualization techniques suggests to us a simple pedagogical policy. Data exploration focused on the number of attempted exercises combined with classification led us to identify students at risk, those who have not trained enough. Clustering and cluster visualisation led us to identify a particular behaviour among failing students, when students try out the logic rules of the pop-up menu of the tool. As in [7], a timely and appropriate warning to students at risk could help preventing failing in the final exam. Therefore it seems to us that data mining has a lot of potential for education, and can

bring a lot of benefits in the form of sensible, easy to implement pedagogical policies as above.

The way we have performed clustering may seem rough, as only few variables, namely the number and type of mistakes, the number of exercises have been used to cluster students in homogeneous groups. This is due to our particular data. All exercises are about formal proofs. Even if they differ in their difficulty, they do not fundamentally differ in the concepts students have to grasp. We have discovered a behaviour rather than particular abilities. In a different context, clustering students to find homogeneous groups regarding skills should take into account answers to a particular set of exercises. Currently, we are doing research work along these lines.

References

[1] Beck, J., ed. *Proceedings of ITS2004 workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes*. Maceio, Brazil (2004).

[2] Mostow, J. "Some Useful Design Tactics for Mining ITS Data" in *Proceedings of ITS2004 workshop Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes*, Maceio, Brazil (2004).

[3] Heiner, C., J. Beck, & J. Mostow. "Lessons on Using ITS Data to Answer Educational Research Questions" in *Proceedings of ITS2004 workshop Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes*, Maceio, Brazil (2004).

[4] Gueraud, V. & J.-M. Cagnat. "Suivi à distance de classe virtuelle active" in *Proceedings of Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et l'Industrie (TICE 2004)*, pp 377-383, UTC Compiègne, France (2004).

[5] Duval, P., A. Merceron, M. Scholl, & L. Wargon. "Empowering learning Objects: an experiment with the Ganesha Platform" in *Proceedings of ED-MEDIA 2005*, Montreal, Canada (2005).

[6] Mazza, R. & V. Dimitrova. "CourseVis: Externalising Student Information to Facilitate Instructors in Distance Learning" in *Proceedings of 11th International Conference on Artificial Intelligence in Education (AIED03)*, F. Verdejo and U. Hoppe (Eds), Sydney: IOS Press (2003).

[7] Minaei-Bidgoli, B., D.A. Kashy, G. Kortemeyer, & W.F. Punch. "Predicting student performance: an application of data mining methods with the educational web-based system LON-CAPA" in *Proceedings of ASEE/IEEE Frontiers in Education Conference*, Boulder, CO: IEEE (2003).

[8] Merceron, A. & K. Yacef. "A Web-based Tutoring Tool with Mining Facilities to Improve Learning and Teaching" in *Proceedings of 11th International Conference on Artificial Intelligence in Education.*, F. Verdejo and U. Hoppe (Eds), pp 201-208, Sydney: IOS Press (2003).

[9] Romero, C., S. Ventura, C. de Castro, W. Hall, & M.H. Ng. "Using Genetic Algorithms for Data Mining in Web-based Educational Hypermedia Systems" in *Proceedings of AH2002 workshop Adaptive Systems for Web-based Education*, Malaga, Spain (2002).

[10] Han, J. & M. Kamber, *Data mining: concepts and techniques*, San Francisco: Morgan Kaufman (2001).

[11] SPSS, *Clementine*, www.spss.com/clementine/ (accessed 2005)

[12] Benchaffai, M., G. Debord, A. Merceron, & K. Yacef. "TADA-Ed, a tool to visualize and mine students' online work" in *Proceedings of International Conference on Computers in Education, (ICCE04)*, B. Collis (Eds), pp 1891-1897, Melbourne, Australia: RMIT (2004).

[13] SODAS, http://www.ceremade.dauphine.fr/~touati/sodas-pagegarde.htm (accessed 2003)

[14] Agrawal, R. & R. Srikant. "Fast Algorithms for Mining Association Rules" in *Proceedings of VLDB*, Santiago, Chile (1994).

[15] Yacef, K., "The Logic-ITA in the classroom: a medium scale experiment". *International Journal on Artificial Intelligence in Education*. 15: p. 41-60 (2005).

[16] Merceron, A. & K. Yacef, "Mining Student Data Captured from a Web-Based Tutoring Tool: Initial Exploration and Results". *Journal of Interactive Learning Research (JILR)*. 15(4): p. 319-346 (2004).

[17] Boud, D., R. Keogh, & D. Walker, eds. *Reflection: Turning Experience into Learning*. Kogan Page: London (1985).