# EXTRACTING MORE MEANING FROM CAA RESULTS THROUGH MACHINE LEARNING

Salvatore Valenti and Alessandro Cucchiarelli

# Extracting More Meaning from CAA Results Using Machine Learning

Salvatore Valenti
Alessandro Cucchiarelli

Istituto di Informatica - University of Ancona
Via Brecce Bianche - 60131 Ancona – Italy

valenti@inform.unian.it

## Abstract

This work describes a novel approach to the problem of extracting knowledge from the results obtained via a CAA system by adopting a Machine Learning paradigm.
The basic idea guiding our research was to investigate the existence of association rules among the topics covered in a course. The data used came from the exams administered to the freshmen in electronic engineering attending the course of Foundation of Computer Science at the University of Ancona. Ten Multiple Choice Questions with four possible answers constituted an exam. Questions have been classified according to the topic covered in a taxonomy derived from the course syllabus. Each question has an absolute weight representing its relative importance inside the curriculum. The data have been filtered by removing low-end and high-end achievers to obtain a subset containing information free from border effects. Each questionnaire has been coded into a vector of features (one for each element of the questions' taxonomy) representing the student's answers (right, wrong, not given). The feature vectors are further classified with respect to the final score obtained by the student (poor, average or good) and analysed using C4.5, a classification system based on top-down induction of decision trees that allows generating production rules.
We classified the generated rules into three categories: "straightforward", "reasonable" and "unexplainable". Rules are considered "straightforward" when they put in relation topics that we believe are related. "Reasonable" rules put in relation topics that although not being predictable by our experience, may be understood after a deeper analysis of the questions. "Un-explainable" rules put in relation topics that do not appear to be related in any way.
A first interesting result of the method discussed is represented by the so-called "reasonable rules" that may be used to better tune the teaching of the topics that appear to be related.

## Introduction

According to Kleeman (2000), “much of the reporting and analysis of Computer Assisted Assessment tends to follow the model for paper tests. The various reports that people used to make for paper tests are duplicated for computer tests. For example, analysis is made of the quality of questions and choices and of the reliability and validity of the test”.

In the quest of novel approaches for analyzing the answers provided by the students, one of the ideas that caught our interest was the possibility of evaluating the existence of correlations among the topics covered by the questions. As our teaching experience in the course of Foundation of Computer Science (Fondamenti di Informatica) has taught us, there is a number of topics that appear to be related. Thus, for instance, we did notice that if a student fails to understand the parameter passing mechanisms usually he/she shows difficulties in implementing recursive algorithms, too. In order to verify this feeling from a more scientific point of view, and to extract more knowledge about the existence of further, less evident, association rules among other topics, we decided to apply data mining techniques to our questionnaire bank.

Thus, this work is aimed to identify the existence of correlations among the formative deficiencies emphasized by the presence of wrong answers to questions related to different topics. If such correlations do exist and can be elicited, a more appropriate feedback than the simple indication of the existence of a wrong answer can be provided to the students. Furthermore, the identification of unexpected or unforeseen correlation among topics may help the teacher to revise the didactic process.

The data used came from the questionnaires administered to the freshmen in electronic engineering attending the course of Foundation of Computer Science at the University of Ancona. A questionnaire was constituted by ten Multiple Choice Questions with four answers, one being the key and the other acting as distractors. Each question has an absolute weight representing its relative importance inside the curriculum. The score obtained by a student ranges from 0 to 30. Questions have been classified according to the topic covered in a taxonomy derived from the course syllabus. The data have been filtered by removing low-end and high-end achievers to obtain a subset containing information free from border effects. Each questionnaire has been coded into a vector of features (one for each element of the questions’ taxonomy) representing the student’s answers (right, wrong, not given). The feature vectors are further classified with respect to the final score obtained by the student (poor, average or good). The feature vectors have been analysed using C4.5, a classification system based on top-down induction of decision trees (Quinlan, 1993) that allows generating decision trees and production rules.

In the following sections we will provide a short description of the C4.5 system and of the data sample used. Then we will discuss some preliminary results that we have obtained so far.

## The C4.5 system

For the purpose of our research, we are interested in Data Mining techniques, i.e. techniques for finding and describing structural patterns in data, as a tool for helping to explain the data and make prediction from it (Witten and Frank,

2000). In data mining, the data take the form of a set of examples and the result takes the form of a prediction on new examples. We are also interested in describing patterns in data, so the output must include an actual description of a structure that can be used to classify unknown examples in order that the decision can be explained.
In our experiments we used the C4.5 package, a set of software tools able to learn inductively models of different concepts (classes) from a set of example, to build a classifier in the form of a decision tree and to use it for unknown example classification.
The inductive method used by C4.5 for classification needs the following key requirements (all satisfied by the experimental asset we use, as described in the next section):

- *Attribute-value description:* the data to be analysed must be structured in a 'flat-file'. All information about one object or case must be expressible in terms of a fixed collection of properties or 'attributes'. Each attribute may have either discrete or numeric values, but the attribute used to describe a case must not vary from one case to another.
- *Predefined classes:* the categories to which cases are to be assigned must have been established beforehand. This qualifies the leaning process used by C4.5 as *supervised*, as contrast with *unsupervised* learning in which appropriate groupings of cases are found by analysis.
- *Discrete classes:* the classes must be sharply delineated (an example either does or does not belong to a particular class) and there must be far more examples than classes.
- *Sufficient data:* inductive generalization proceeds by identifying patterns in data. The approach fails if valid, robust patterns cannot be distinguished from chance coincidences. As this differentiation usually depends on statistical tests of one kind or another, there must be sufficient cases to allow these tests to be effective.
- "*Logical" classification models:* the programs construct only classifiers that can be expressed as decision trees or sets of production rules. These forms essentially restrict the description of a class to a logical expression whose primitives are statements about the values of particular attributes.

The program generates a classifier in the form of a decision tree, a structure that is either:

- a *leaf,* indicating a class, or
- a *decision node* that specifies some test to be carried out on a single attribute value, with one branch and sub tree for each possible outcome of the test.

A decision tree can be used to classify a case starting at the root of the tree and moving through it until a leaf is encountered. At each nonleaf decision node, the case's outcome for the test at the node is determined and attention shifts to the root of the sub tree corresponding to this outcome. When this process finally leads to a leaf, the class of the case is predicted to be that recorded at the leaf.
Constructing a classification model is not limited to the development of accurate predictors: another principal aim is that the model should be intelligible to human beings, so that they can deepen the comprehension of the domain described by the cases. To achieve this goal C4.5 is able to re-express a classification tree as production rules, a format that appears to be

more intelligible than trees. The program uses a simplified form of production rule *L -> R* in which the left-hand side *L* is a conjunction of attribute-based tests and the right-hand side *R* is a class.
Along with the induced tree (or rules, if the corresponding option is selected) each invocation of the classifier over a set of cases produces an output containing its performance on the cases from which it was constructed. In other words, after classes induction, the program applies the classification to the same cases used for training, and evaluates the percentage of errors made: the less this error is, the better the model of classes is.
The program also contains heuristic methods for simplifying decision tree, with the aim of producing more comprehensible structures without compromising accuracy on unseen cases. Both tree (or rules) and error values related to the simplified tree are printed to the output stream.
The method commonly used for estimating the reliability of a classification model is to divide the data into a training and test set, build the model using only the training set, and examine its performance on the unseen test cases. This is quite satisfactory when there is plenty of data, but in the more common circumstances of having less data than we would like, two problems arise. First, in order to get a reasonably accurate fix on error rate, the test must be large, so the training set is impoverished. Secondly, when the total amount of data is moderate, different divisions of the data into training and test set can produce surprisingly large variations in error rates on unseen cases.
A more robust estimate of accuracy on unseen cases can be obtained by cross-validation. In this procedure, the available data is divided into N blocks so as to make each block's number of cases and class distribution as uniform as possible. N different classification models are the built, in each of which one block is omitted from the training data, and the resulting model is tested on the cases in that omitted block: In this way, each case appears in exactly one test set. Provided that N is not too small (10 is a common value) the average error rate over the N unseen test set is a good predictor of the error rate of a model built from all the data. The C4.5 package contains a module that automates the evaluation of accuracy of a classification model through cross-validation.

## Description of the Data Sample

The data used for knowledge discovery came from the exams administered to the freshmen in electronic engineering attending the course of Foundation of Computer Science during the years comprised between 1995 and 1998. An exam was constituted by ten Multiple Choice Questions with four answers, one being the key and the other acting as distractors. Each question was given a weight ranging from 1 to 5 points, representing the relative importance of the question inside the curriculum. The number of different questions in the database is 150. The questions have been validated and verified more than once according to the classical approaches of Item Analysis. Thus, the questions have been blind-reviewed at different times by different experts who had to rate the item-to-content congruence. Furthermore, all the questions were evaluated according to basic statistical approaches by calculating the Proportion Correct Index (p-value), and the Item Discrimination Indexes as for instance the point biserial estimate of correlation, the biserial correlation coefficient and the phi correlation estimate (Osterlind, 1998).

Questions have been classified according to the topic covered, in a 17 items taxonomy, derived from the course syllabus, and listed in table 1.

| Topic | Questions Argument |
|---|---|
| Integer and Real data type | Use of simple types and variables – internal representation |
| Operating System | Shell commands |
| Grammars & EBNF | Grammar definitions and use of EBNF |
| Operators and Types | Use and definition of operators and types |
| Binding | Binding methods |
| Arrays | Definition, representation and use and of arrays |
| Lists | Definition, representation and use and of lists |
| Trees | Definition, representation and use and of trees |
| Files | Definition, representation and use and of files |
| Tables | Definition, representation and use and of tables |
| Selection constructs | Examples of Selection constructs |
| Iterative constructs | Examples of Iterative constructs |
| Scope of variables | Mechanisms of variables scoping |
| Parameters | Techniques for parameter passing |
| Side effects | Procedures and functions with side effects |
| Recursion | Recursive algorithms |
| Programming Language | Pascal |

**Table 1 – Topics covered by the questions used to assess the students**

Each questionnaire contains exactly two questions for each weight. The minimum score that may be obtained by a student is zero and the maximum is 30.
We started with a database of 1322 questionnaires. The data has been filtered by removing low-end and high-end achievers in order to obtain a subset containing information free from border effects. We decided to analyse questionnaires containing 3 or 4 wrong answers, thus taking into account only those whose final score was greater then 12 and lower than 26. The rationale for selecting this threshold is due to the consideration that questionnaires with a greatest number of wrong answers contain too many errors and are usually the result of a poor level of study. Questionnaires with a lower level of wrong answers produce data that is difficult to investigate since they contain too few errors, and so they add very little significance to our research. After this filtering, the data set obtained contains 436 questionnaires whose distribution of grades is represented in fig. 1.

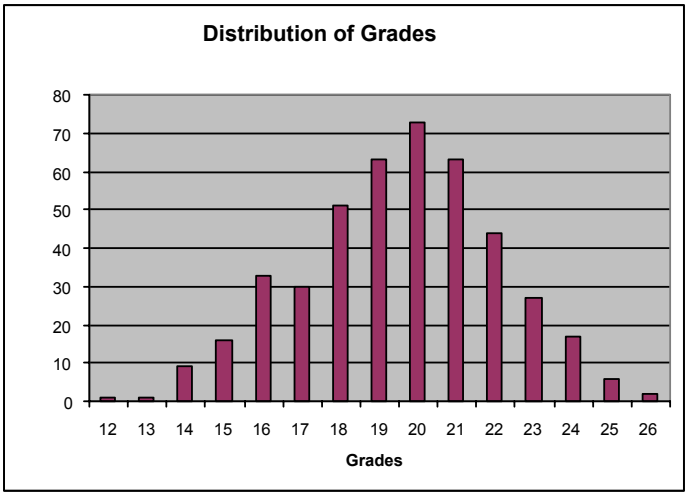


**Figure 1 – Distribution of grades in the dataset**

Each questionnaire is represented as in table 2:

| **Student Id** = 123 | **Date** = 19/06/98 | **Score** = 15 |
|---|---|---|
| **Question** | **Topic** | **Answer** |
| 1 | Integer and Real data type | wrong/missing |
| 2 | Grammars & EBNF | correct |
| 3 | Arrays | wrong/missing |
| 4 | Operators and Types | wrong/missing |
| 5 | Lists | correct |
| 6 | Grammars & EBNF | correct |
| 7 | Selection constructs | correct |
| 8 | Parameters | wrong/missing |
| 9 | Trees | correct |
| 10 | Arrays | correct |

**Table 2 – A sample questionnaire**

The table is transformed in a set of feature vectors:

$$A_1, A_2, A_3, \ldots, A_{16}, A_{17}, P$$

where:

- $A_i$ is 0 if no question addressing topic i is present in the questionnaire, 1 if the question containing the topic i is correctly answered, 2 if the answer to the question containing topic i is wrong or missing;
- P is “poor” if the score to the questionnaire is between 12 and 17, “average” if the score of the questionnaire is between 18 and 20, and “good” if the score is between 21 and 26.

Since the same topic may be covered by different questions in each questionnaire, the example in table 2 is transformed in the following set of feature vectors:

| |
|---|
| 2, 0, 1, 2, 0, 2, 1, 1, 0, 0, 1, 0, 0, 2, 0, 0, 0, poor. |
| 2, 0, 1, 2, 0, 1, 1, 1, 0, 0, 1, 0, 0, 2, 0, 0, 0, poor. |
| 2, 0, 1, 2, 0, 2, 1, 1, 0, 0, 1, 0, 0, 2, 0, 0, 0, poor. |
| 2, 0, 1, 2, 0, 1, 1, 1, 0, 0, 1, 0, 0, 2, 0, 0, 0, poor. |

This way, the 436 questionnaires allowed generating 7884 feature vectors. The analysis of the features vectors with the C4.5 program allowed constructing a decision tree that classifies the questionnaires under examination with an error rate of 11.9%. Then a list of about 201 production rules was generated from the decision tree. The rules allow to classify the behaviour of the student according to the score classes defined (poor, average and good) with an overall error of 13.4%. Thus, for instance, the following rule has been generated:

Scoping of variables = 2
Parameters = 2
-> class “poor” [98.7%]

The rule allows to infer that if a student does not answer correctly to questions regarding the topics: “Scoping of Variables” and to “Parameters” is ranked “poor” (i.e. his/hers questionnaire has a score between 12 and 17) with a confidence factor of 98.7%. Thus, the two topics appear to be correlated. This remark is further strengthened by the existence of the following rule:

Scoping of variables = 1
Parameters = 1
-> class “good” [99.8%]

According to this rule, if a student answers correctly to questions regarding to the same topics his/hers questionnaire will obtain a score comprised between 21 and 26 with a confidence factor of 99.8%. Thus it becomes evident that the two topics are strongly related.

In order to get a more robust estimate of the accuracy of the model devised, we decided to apply the cross-validation utility provided by the C4.5 system as described in the previous section. We set N=10, so obtaining very encouraging results since the average error rate is 15,9% if the decision tree is applied, 16.2% if the rules are used.

## Preliminary Results

We decided to classify the rules generated by C4.5 into three categories: “straightforward”, “reasonable” and “unexplainable”.
Rules are classified as “straightforward” when they put in relation topics that we believe are associable. The rules above represent two examples of straightforward associations, since we strongly believe that a student failing to catch the concept of “Scoping of variables” will have difficulties in understanding the correct use of “Parameters” (and vice versa).

Reasonable rules are those that put in relation topics that although not being predictable by our experience, may be understood after a deeper analysis of the questions. Thus, for instance, the rule:

```
Grammars & EBNF = 2
        Side Effects = 2
Recursion = 2
-> class “poor”  [96.8%]
```

allows to conclude that a student is ranked “poor” with a confidence factor of 96.6% if his/hers questionnaire contains wrong answers to questions related to Grammars and EBNF, to Side Effects and to Recursion. While we believe that it is possible to put in relation the use of functions containing Side Effects and Recursion, so that a misconception on each of the topics may affect the other, it appears difficult to understand in which way the topic Grammars & EBNF should be put in relation with the formers. But, after a careful analysis, we discovered that most of the questions related to this latter topic contain excerpts of production rules that often involve recursive definitions. Thus, the conclusion that a student failing to understand the concept of recursion may have difficulties in answering questions involving recursive production rules does not appear as an extremely odd idea.

“Un-explainable” rules are those who put in relation topics that do not appear to be related each other in any way. Thus, it is very hard to understand the meaning of the following rule:

```
Integer & Real data type = 1
Side Effects = 1
-> class “good”  [96.1%]
```

that may allow inferring that if a student answers correctly both to a question related to the use of Integers and Reals and to a question covering the topic Side Effects, is ranked “good” with a confidence factor of 96.1%.
We believe that this last category of rules deserves a deeper analysis, since it may be useful to obtain a better understanding of the way students perform.

## Final Remarks

In this paper we described a novel approach for the analysis of the questionnaires provided by the students, in order to identify possible relations among topics. The approach is based on the use of the C4.5 package that allows the extraction of decision trees and production rules from a data set. The data set was constituted by 432 questionnaires used as final exams for the course of Foundation of Computer Science at the University of Ancona. The approach seems promising since it allowed identifying three categories of production rules that may be used to classify relations among the topics covered by the questions as Straightforward, Reasonable, and Unexplainable. This could represent a first step towards the construction of a predictive model for the evaluation of the students’ learning.

Both the last two categories of rules require further investigation. One point that remains to be addressed so far is to demonstrate that the Unexplainable rules are not due to some sort of statistical error. As a first step in this direction, the use of the cross validation utility provided by the C4.5 package seems to indicate that the obtained results are reasonably stable in a way that appears to be independent from the data set used.

## References

Gleeman, J. (2000). Getting More Meaning from the Results of CAA – Discussion forum introduction **in** Danson M., and, Hilton A. (eds.), Proceedings of the 4th International CAA Conference, Loughborough, UK: Loughborough University.

Osterlind, S.J. (1998). Constructing Test Items. 2nd ed. Boston: Kluwer Academic Publishers, 1998.

Quinlan, J.R. (1993). C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann Publishers Inc.

Witten, I. H, Frank, E. (2000), Data Mining, San Mateo, CA: Morgan Kaufmann Publishers Inc.