Practical Applications of Intelligent Systems: An E-Learning Platforms Recommender

Victoria Lopez, Matilde Santos
Dept. Computer Architecture and Automatics
Complutense University
Madrid, Spain
vlopez@fdi.ucm.es

Pablo Esteve, David Roldán, Juan C. deMendoza
Informatics Faculty
Complutense University
Madrid, Spain

Abstract—**This paper shows the advantage of using Intelligent Systems while developing engineering projects. The paper is based on three examples and it is focus into one of them: The E-learning Migration application, developed as a teaching aid tool in different organizations where various platforms coexist; this is the case of some Spanish universities. This tool solves three basic problems, which are shown in three independent modules. The first of these ones, concerns the recording and migration of the data; the second one, refers to the storing and organization of the images. The last one is an E-learning platform recommender based on an ID3 tree-decision. We are going to focus our article in this concrete module.**

Keywords-E-Learning; ID3, recommender systems, intelligent systems.

I. INTRODUCTION

There are a lot of applications developed as engineering projects in which meet natural language and quantitative data. In theses cases, Intelligent Systems offers a set of suitable techniques and very easy to apply. The following three examples were developed at Complutense University of Madrid during this academic course.

- SB Data Analysis [11]: A study about Store Brands (SB) and market consumers is developed in this project. Fuzzy logic and Multicriteria Decision Making techniques are used to decide the behavior of the market. The software tool SB Data Analysis allows make several store brand surveys analysis and make a comparison of store brands' consume by age group and waves using Soft-Computing techniques. These studies will serve to understand better the evolution, trends and tastes of consumers with respect to store brands.
- EMSI [10]: Measuring computer performance and reliability. This project develops some tools and techniques in computer-system performance and reliability analysis. Statistical methods combined with fuzzy logic techniques are used to interpret the measured data in a meaningful way. Also in this project the user can find an algorithm (following a pseudo greedy scheme) to improve reliability and performance of a set of devices or components into a computer-system. In particular, a fuzzy multicriteria decision making (MCDM) approach is considered at each step of the algorithm to select a new candidate, so the greedy scheme improves each isolated component. At the final stage, a new system is obtained and its global reliability and performance can be then evaluated by means of appropriate fuzzy specifications.
- E-Learning Management Tool [8]: This project has got three modules, one of them allows to migrate contents that a teacher has in a platform to another one, the second one allows to store in a database images allowing the teacher to annotate them with Topics or descriptions that he chooses, as well as to assign to the images subjects, so that then they could select images for a concrete subject or description. In addition the export is allowed in a compressed file of the images that the user chooses. The third tool is a system recommender for learning platforms. This recommender are based on artificial intelligent technique, the ID3 algorithm that it is shown into next sections.

II. E-LEARNING MANAGEMENT TOOL

The increasing popularity of open source and freeware platforms added with the traditional use of commercial platforms, point to the benefit of migrating data between them with the most transparency and reliability for the user.

During the last years, there has been a steady increase in online educational systems, also known as E-learning systems. Nowadays is not very frequent to find educational centers, or even companies who are not in the possession of a 'Learning Management System' (LMS).

At the beginning, these systems were for commercial reasons like WebCT (Blackboard as the new version) [1] however, freeware systems such as Moodle [6] are currently playing a major role. With regard to this topic there are a number of articles that can be consulted [5].

As the use of such virtual learning systems keeps increasing, the E-learning Migration tool arises to simplify the user's work. On the one hand, our application seeks to make

978-1-4244-6793-8/10/$26.00 ©2010 IEEE

the migration of groups of questions easier, which the user ought to have stored in one concrete platform (questions bank), between other platforms that are being used currently like: WebCT [1], Moodle [6], Sakai [9] and Ilias [4]. In addition, the questions can be created, erased and changed with the advantage of been independent from any platform, for the latter exportation to any of them. The first figure shows this module. Another of the utilities of this application is the management of an image data base that will simplify the incorporation of these images in any of the platforms named previously. Moreover, this utility works as an image repository grouping them by topics according to the user's profile. These different utilities achieve their aim of increasing the teacher's (or a general user) productivity when it comes to making a presentation or any other document. The second figure shows the images repository.

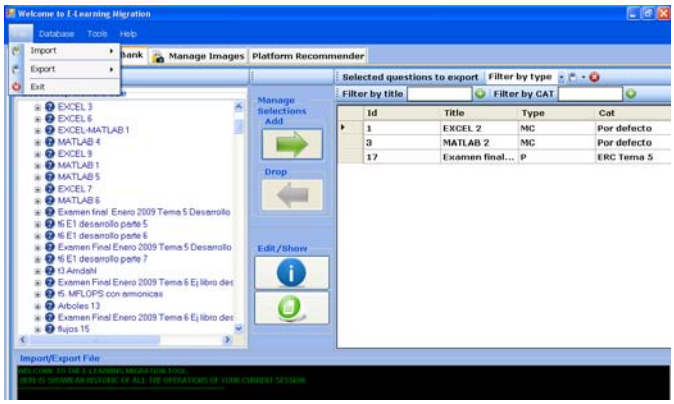

Figure 1. Migration Module

The users of virtual integrated campus are used to encounter different problems when they are making the decision of which platform they should use. Users can feel overwhelmed both with the wide range of available platforms and the scanty knowledge they have to decide which platform best fits in each case.

Figure 2. Images repository module

This way comes up the third utility of the tool under consideration in this article: a recommender system, based in a *ID3 classification* algorithm, that automate the selection of the platform, taking into account the user profile and his needs.

Section 2 is a presentation about the ID3 algorithm. The result of applying the algorithm to the recommender of platforms will be described in Section 3, including a sample set of rules used, the decision tree and other results obtained. Finally, Section 4 is devoted to conclusions and future work.

III. ID3 ALGORITHM

Developed by J. Ross Quinlan in 1983 [2,3], the algorithm ID3 is used within the field of artificial intelligence. Its use is included in the search of hypotheses or rules, given a set of examples that consists of a series of values. Each of these sets of values or tuples are attributes, and one of them (the attribute to classify) is the objective, which is binary type (positive or negative, yes or not, valid or invalid, etc.). In this particular application, the classifier attribute has four possible values, one for each possible platform recommended (WebCT [1], Moodle [6], Sakai [9] and Ilias [4]). The algorithm seeks to rank the scenarios, with new bodies, if this example will be positive or negative, or equivalently, what is the recommended platform.

ID3 does this by building a decision tree. The elements that make up the decision trees are explained below:

- Node: Names or identifiers of the attributes.
- Arches: Possible attribute values associated with the node.
- Leaves: Sets and examples classified and labeled with the name of a class.

The objective is to choose the best attribute according to a given heuristic. This is essential to identify the variables that provide information relevant to the classification of the solution. This solution consists of a sequence or path of the decision tree. Using the ID3 algorithm is interesting for this purpose because of their characteristics, among which include the following:

- The goal is to create a decision tree as a method to approximate an objective function of discrete values, which is resistant to noise in the data and be able to find or learn of a disjunction of expressions.
- The result can be expressed as a set of if-then rules.
- Try to find the easier tree that separates better examples.
- It is recursive.
- Does not use "backtracking".
- Uses the concept of entropy.

Among the components of the ID3 algorithm it is important to define the following items:

- Attributes: These are the factors that influence the classification or decision. The selection of attributes is based on the experience already

gained. In the algorithm, each attribute is an intermediate node of the tree, where leaves are the lessons or decisions.

- Class: Possible values of the solution.
- Examples: The set of combinations of attribute values given as a training set. Given this training set, the algorithm selects the attribute that divides the examples by the best way basing on the concept of entropy, which is explained below.
- Entropy: It is the way to measure the uncertainty of the system. Definitely, it shows us, in a particular situation, the probability of each possible outcome. In the background is only a heuristic, which as we shall see, will help us choosing the best attribute at each node
- Choosing the best attribute: To elect the most discriminating attribute in each case the tree uses the concept of entropy. The selection criterion chooses the attribute that provide better information gain. The function chosen may vary, but in its simplest form is:

$$-\left(\frac{|p|}{|d|}\right)\log_2\left(\frac{|p|}{|d|}\right)-\left(\frac{|n|}{|d|}\right)\log_2\left(\frac{|n|}{|d|}\right)$$

Where 'p' is the set of positive examples, 'n' the negative one, and 'd' all of them.

- ID3 examines all attributes and choose the one that provide maximum gain, build a branch and uses the same process recursively to form sub-trees from the 'v' nodes generated.

Figure 3 shows the algorithmic scheme for the construction of ID3 decision trees.

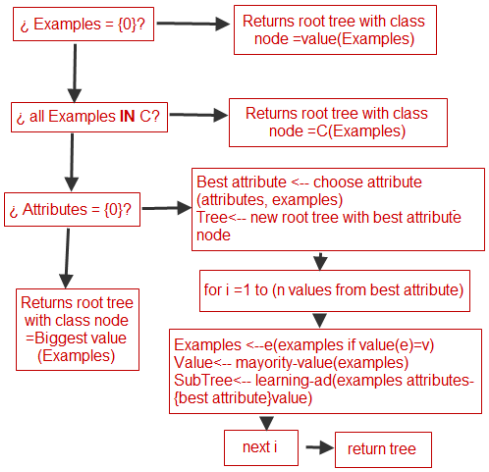

Figure 3. Algorithmic scheme for the ID3 algorithm

In order to show more easily the performance of the algorithm, we include a typical example. The example is based on the decision problem that arises about playing tennis or not depending on environmental conditions. This takes into account several attributes (atmosphere, temperature, humidity and wind) reported with their corresponding tuples of values below:

- Atmosphere: Sunny-Cloudy-Rain
- Temperature: High-Low-Soft
- Humidity: High-Normal
- Wind: Strong-Weak
- Result: Play-Rest

Table 1 below shows some values for N days taken as examples to feed the algorithm.

TABLE I. EXAMPLES AND ATTRIBUTES

Ejem.	Atm.	Temp.	Hum.	Wind	Res.
1	Sunny	High	High	Strong	Rest
2	Rain	Soft	High	Weak	Play
3	Rain	Low	Normal	Strong	Rest
4	Sunny	Soft	High	Weak	Rest
5	Rain	Soft	Normal	Weak	Play
6	Cloudy	Soft	High	Strong	Play
7	Rain	Soft	High	Strong	Rest
…	…	…	…	…	…

In this case the algorithm ID3 help to classify the result attribute with possible values {play or rest}, depending on the attributes just mentioned. From these data, generates the decision tree shown in Figure 4.

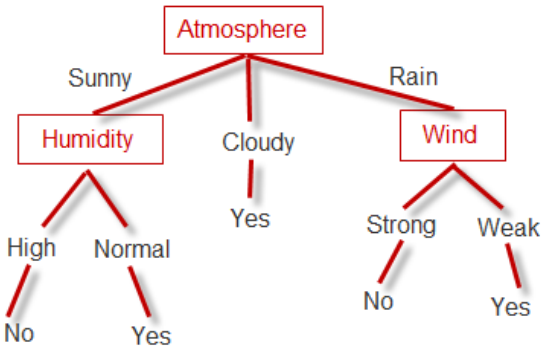

Figure 4. Tree Decision

Based on information obtained from the decision tree, we obtain the classification rules. For example if it is sunny and humidity is normal, play tennis will be positive.

The concept of entropy helps to ignore attributes that are not useful when making a decision, as it would be -in this case- the temperature attribute, which it is not necessary to classify the result. This concept substantially improves the result and execution time.

Algorithm 1 shown below presents the pseudo-code that builds ID3 decision tree.

```
Procedure Id3 (Input: Examples, Attribute-
goal, Attributes; Output: Attribute goal (+,-
))

If all examples are positive then return a
positive node +

End

If all examples are negative then return a
negative node –

End

If Attributes=∅  return majority vote from
goal-attribute value in Examples

End

Else,

  Be A the BEST attribute

  Foreach v values of the attributes do

      Be Examples (v) the subset of examples
who's A attribute value is v

      If Examples (v) is  empty return a node
with majority vote from goal-attribute value
in Examples

      else return Id3(examples(v), Attribute-
goal, Attributes/{A})
```

IV. E-LEARNING PLATFORMS RECOMMENDER

This section focuses on the application of the ID3 algorithm to the recommendation of a learning platform.

A. Selection of attributes:

To the recommender of platforms we have selected the attributes whose values serve as training sets. These attributes are:

- Type of the question: It takes into account the types of questions the user wants to include, as not all platforms can work with all types of questions. The question types supported by each of the platforms are listed in Table 2, shown below:

TABLE II. QUESTION TYPES SUPPORTED BY EACH PLATFORM

	MC	S	P	C	M	NR	FB
WebCT	X	X	X	X	X		
Moodle	X	X			X	X	X
Sakai	X	X			X	X	X
Ilias	X	X			X	X	X

The meanings of the symbols of the types of questions are:

- MC: Multiple choice
- S: Short answer
- P: Long answers
- C: Calculated answers based on a formula
- M: Matching pairs
- NR: Numeric responses
- FB: Fill in blanks

From these platforms we can see that all of them support the same number of types but the ones that have more in common are Sakai, Ilias and Moodle; while WebCT has rates questions that only support this platform.

- **Images:** The user must assess whether to require the use of images on his platform, because not all platforms are supported. The possible values are (yes / no).
- **Interface:** Another attribute to take into account will be how complex the recommender interface is, as someone may want a simple interface and less eye-catching, while other users may prefer more complicated interfaces but more complete. Therefore the possible values are (Simple / complex).
- **Price:** In order to choose which E-Learning platform use, a lot of people worry about money -especially these days- so another option that will have the user is using a free or paid platform.
  - Free Platforms:
    - Sakai
    - Ilias
    - Moodle
  - Pay Platforms :
    - WebCT
- **User knowledge:** Also gives users the option of choosing any of the existing platforms if they have previously worked with any of them so that the recommender will take this into account when classifying the result.
- **File format import:** This attribute can be critical for some users, as there are users who do not know XML and probably prefer to work with text files.
  - **Format types:**
    - XML
    - TXT

B. Tree generated with ID3

By the previous training set and using the algorithm shown in section 2 of this article, we get the ID3 decision tree that enabes obtaining the decision rules to classify and recommend the user a specific platform. Figure 5 shows the tree structure generated.

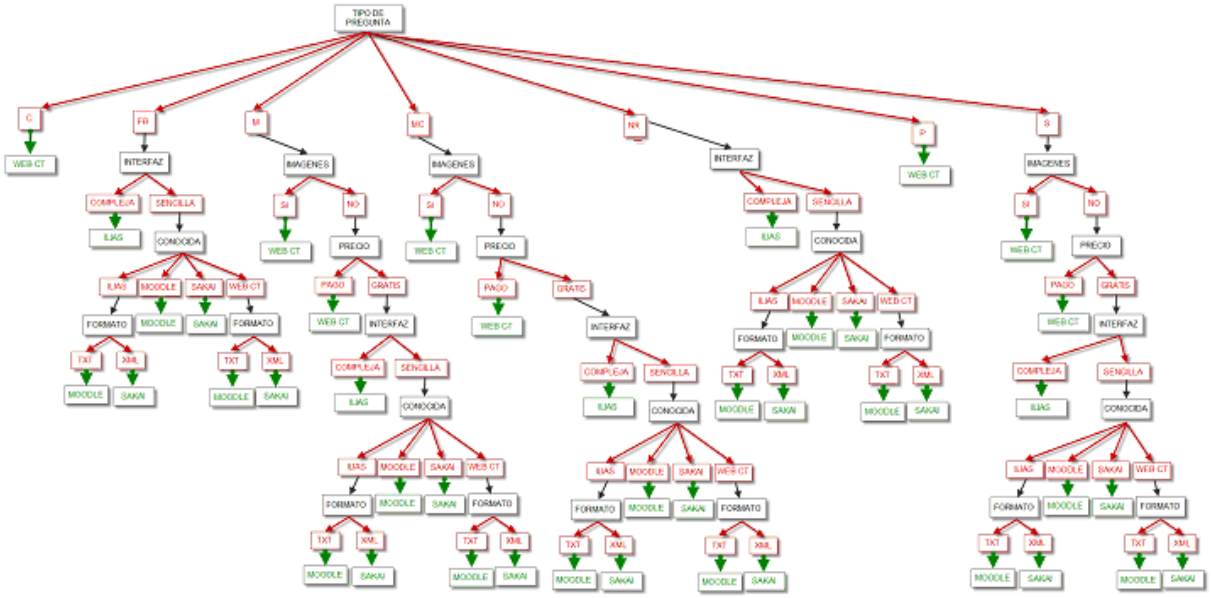

Figure 5. ID3 Tree

C. *Obtained Rules:*

From the previous decision tree we can obtain the rules that will use the recommender. Figure 6 shows some of the rules used to illustrate this section. The complete set consists of 43 rules and can be found in [8].

- If "QUESTION TYPE"="M" AND "IMAGES" = "NO" AND "PRICE"="FREE" Y "INTERFACE" = "SIMPLE" AND "KNOWN"="WEBCT" AND "FILE FORMAT"="TXT" -----> "MOODLE"
- If "QUESTION TYPE"="M" AND "IMAGES" = "NO" AND "PRICE"=" FREE" Y "INTERFACE" = "SIMPLE" AND "KNOWN"="WEBCT" AND "FILE FORMAT"="XML" -----> "SAKAI"
- If "QUESTION TYPE"="MC" AND "IMAGES" = "SI" -----> "WEBCT"
- If "QUESTION TYPE"="MC" AND "IMAGES" = "NO" AND "PRICE"="PAY" -----> "WEBCT"
- If "QUESTION TYPE"="MC" AND "IMAGES" = "NO" AND "PRICE"=" FREE" Y "INTERFACE" = "COMPLEX" -----> "ILIAS"

Figure 6. Decision rules

Figure 7 shows the interface with the recommender results.

V. CONCLUSIONS

The ID3 algorithm allows us to obtain a decision tree and a set of rules that enable to obtain a recommender of E-Learning platforms for users who want to use one in concrete with no need of having knowledge of any of them.

After analyzing other artificial intelligence algorithms such as vector quantization or self-organizing maps, and soft computing algorithms (see [7], for example) as blurred or fuzzy logic we chose decision trees and ID3 specifically because it suited better to our propose.

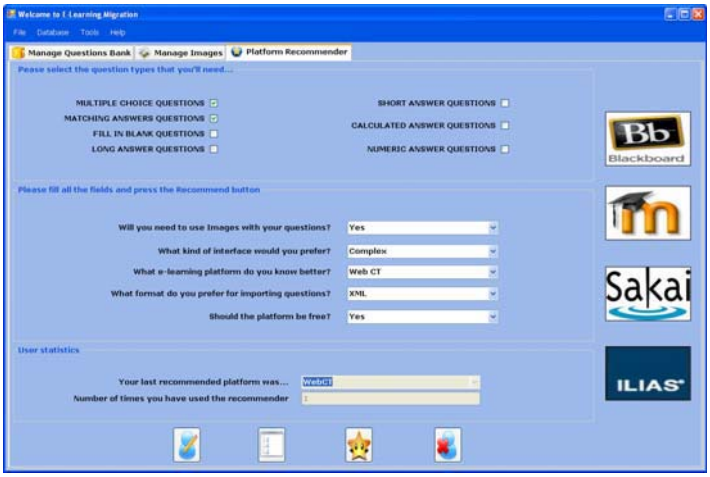

Figure 7. Recommender results

For possible improvements in the recommender in the future, it would be possible to expand the number of platforms that are taken into account for the recommendation, currently limited to 4. It can also be increased the number of attributes whose values are considered by the recommender, valuing factors such as, for example, the number of questions stored in the database of your application, or the time-consuming in an import of files in each of the different platforms. Moreover, studying the characteristics of different users and platforms could give the opportunity of developing a collaborative recommender of platforms.

Acknowledgements

We appreciate the support and funding of CAM research group and project GR58/08 TIN2009-07 901.

REFERENCES

[1] http://www.blackboard.com/

[2] http://www.cis.temple.edu/~ingargio/cis587/readings/id3-c45.html

[3] http://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm

[4] http://www.ilias.de/docu/

[5] V. López, M. Santos y J. A. Martín-H, "Developing computacional tools for the portability of information in different virtual systems," International Technology, Education and Development Conference. Valencia, España, 8-10/03/2010, pp: 5434-5452, 2010

[6] www.moodle.org

[7] G. Pajares y M. Santos Peñas, Inteligencia artificial e ingeniería del conocimiento. Editorial RA-MA, 2005

[8] D. Roldán, P. Esteve, J. de Mendoza, Computer System Project 'E-learning Migration', for publishing, 2010, Informatic Faculty, Complutense University, Madrid

[9] http://sakaiproject.org/

[10] M. Alvarez, C. Padellano, J. de Lara, Computer System Project 'EMSI', for publishing, 2010, Informatic Faculty, Complutense University, Madrid

[11] R. Ortega, P. Jimeno, J. Caballero, Computer System Project 'Store Brands Data Analysis', for publishing, 2010, Informatic Faculty, Complutense University, Madrid