AI Magazine Volume 15 Number 3 (1994) (© AAAI)

Applying Metrics to Machine-Learning Tools

A Knowledge Engineering Approach

Fernando Alonso, Luis Maté, Natalia Juristo, Pedro L. Muñoz, and Juan Pazos

■ The field of knowledge engineering has been one of the most visible successes of AI to date. Knowledge acquisition is the main bottleneck in the knowledge engineer's work. Machine-learning tools have contributed positively to the process of trying to eliminate or open up this bottleneck, but how do we know whether the field is progressing? How can we determine the progress made in any of its branches? How can we be sure of an advance and take advantage of it?

This article proposes a benchmark as a classificatory, comparative, and metric criterion for machine-learning tools. The benchmark centers on the knowledge engineering viewpoint, covering some of the characteristics the knowledge engineer wants to find in a machine-learning tool. The proposed model has been applied to a set of machine-learning tools, comparing expected and obtained results. Experimentation validated the model and led to interesting results.

Learning from examples is currently one of the most active AI research areas (Bratko and Lavrac 1987; Muñoz 1991). Basically, it is important because the results contribute positively to eliminating the bottleneck that knowledge acquisition constitutes in expert system building. As a result, there is a good collection of machine-learning tools for use in knowledge acquisition (Nuñez 1991; Wielinga et al. 1990; Bratko and Lavrac 1987). This proliferation of learning projects and, therefore, of techniques and tools has led to even more interest in comparative studies of such techniques and tools (Kodratoff and Michalski 1990).

This article focuses on a comparative study of how well machine-learning tools deal with ordinal, numeric, and structured attributes as well as cost. It elaborates the metrics to be applied to machine-learning tools, the introduction of these metrics into a model (a benchmark) for application, and concentration on the knowledge engineer as the user of the proposed model.

The main concepts used throughout this article are briefly explained in the next section. Then, a view of different comparative studies is given. Subsequently, the benchmark and some experiments with it are presented. Finally, the conclusions reached are summarized, and some future research directions are given.

Concepts

The development of *expert systems,* that is, programs that are skillful in a particular application domain, emphasized the importance of large stores of domain-specific knowledge (called *knowledge bases*) as a basis for high performance. Assembling and modifying the required knowledge base is a complex process that requires much expertise and careful maintenance.

A key element of this process is the transfer of expertise from a human expert to the program, that is, the task of knowledge acquisition. Because the expert generally knows little about programming, this process usually requires the mediation of a person called a *knowledge engineer.* However, this transfer of knowledge through the knowledge engineer is somewhat problematic. First, the knowledge engineer is not an expert in the specific application domain. Second, because most of the expert knowledge is heuristic and experimental, the expert is not capable of conveying it directly to the knowledge engineer. The field that studies these problems and their possible solutions is called *knowledge engineering.*

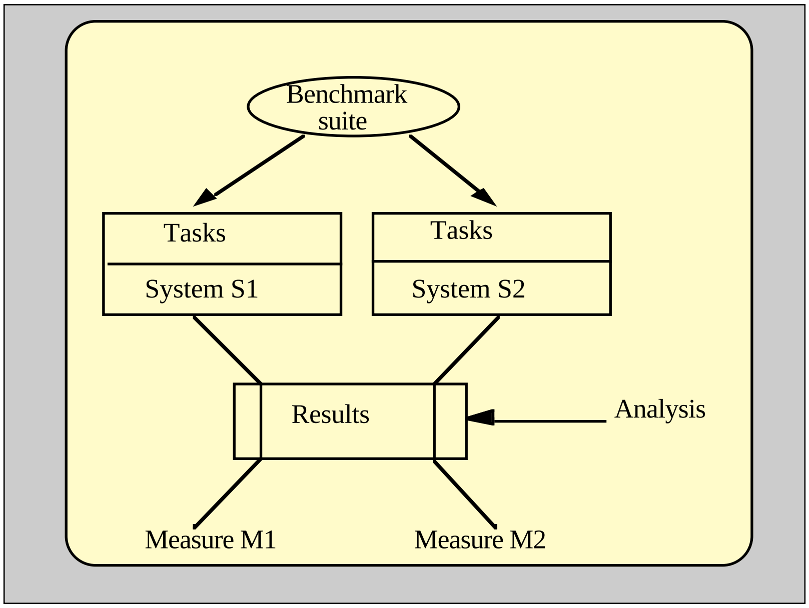

Figure 1. The Benchmark Paradigm.

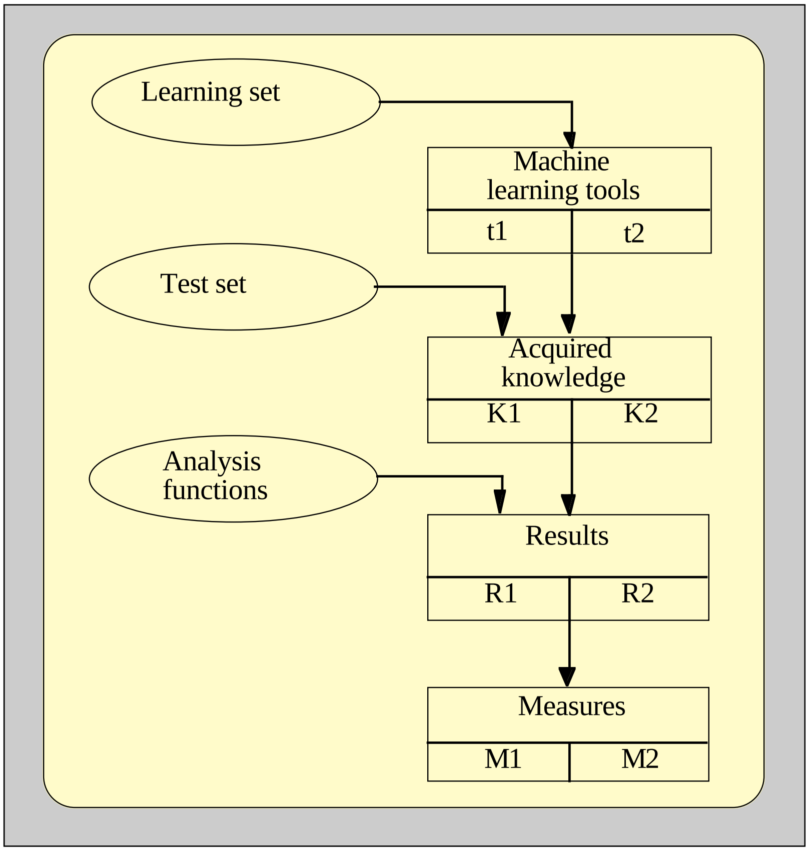

Figure 2. Benchmark Application to Machine-Learning Tools.

Once the software product (for example, an expert system) has been built, it needs to be tested. A *benchmark* can be defined (Alonso et al. 1990) as a technique to evaluate and select software that has already been developed, that is, a software product. Benchmarks are a mixture of jobs, activities, or tasks that are processed by the software to be compared. Three main components are associated with a benchmark (Hayes-Roth 1989): (1) the things or characteristics that the user wants to know about when using the benchmark (*target characteristics*), (2) the tasks to be performed by the target software (*benchmark suite*), and (3) the way in which results are analyzed (*analysis functions*). Figure 1 shows the combination of these three elements.

Figure 1 shows how a system performs a set of tasks as specified by a benchmark suite. These tasks have been defined taking into account the characteristics previously pointed out. The results are then analyzed in the way specified by the benchmark analysis functions. Typically, two (or more) alternative systems, S1 and S2, are subjected to the same experiment to arrive at results, M1 and M2, respectively. Generally, S1 is preferred to S2 if M1 surpasses M2.

Machine learning is a general term denoting the way in which computers increase their knowledge and improve their skills. The field of machine learning studies computational methods for acquiring new knowledge, new skills, and new ways to organize existing knowledge. Machine learning from examples is a specific area of machine learning and can be defined as follows:

> Given a collection of examples that represent a set of concepts to be learned, find a generic description of these concepts for differentiation (Muñoz 1991, p. 58).

Building a benchmark for machine-learning tools involves splitting the benchmark suite into two sets. The first one is the *learning set*. It contains the initial examples that are given to the tools for learning purposes. The second one is the *test set* and contains the examples used for testing the acquired knowledge. Once this knowledge has been tested, the analysis functions are applied to the results and produce the final measures. This process is illustrated in figure 2, which shows the interaction among all these elements.

Related Work

The need for measures is nothing new in learning from examples (Michalski, Carbonell, and Mitchell 1983; O'Rorke 1982), and such measures have centered on different magnitudes. Table 1 shows some of the studies that originated from this need. The first column contains the type of study and the second some references for further information. For an overview of these papers, see Muñoz (1991).

It is evident to those of us in this field that the early studies tend to concentrate on the developer's viewpoint and later ones on the user's viewpoint. Also evident is the lack of any benchmark. These two considerations have been the major inspiration for the research presented here.

TYPE OF STUDY	AUTHOR	YEAR
Comparative study of ID & AQ families	O'Rorke	1982
Comparative review of learning from examples techniques: computational efficiency & others	Dietterich & Michalski	1983
Algorithm complexity	Utgoff	1989
Classification accuracy	Michalski, Mozetic, Hong & Lavrac	1986
	Mingers	1989a
	Quinlan	1989
	Utgoff	1989
Noisy and incomplete data	Cestnik	1987
	Quinlan	1986 & 1989
	Clark & Niblett	1989
Use of common sets of examples for different studies	Cestnik	1987
	Clark & Niblett	1987 & 1989
	Michalski	1990
Obtained results legibility	Cendrowska	1988
	Cestnik	1989
Frameworks for tools' studies	Dhaliwal & Benbasat	1990
	Gams & Lavrac	1987
Comparing Symbolic and Neural Learning	Fisher & McKusik	1989
	Mooney, Shavlik, Towell & Grove	1989
	Atlas, Cole, Connor, Sharkawi, Mars Muthsuamy & Barnard	1990
	Shavlik, Mooney & Towell	1991

Table 1. Comparative Studies of Learning from Examples.

Benchmark Specification

As we said earlier, three components must be specified to define a benchmark: the target characteristics, the benchmark suite, and the analysis functions. A benchmark has been defined for some of the characteristics that a knowledge engineer wants to find in machine-learning tools: ordinal, numeric, and structured attributes as well as economic considerations. The first step in benchmark specification is to identify the target characteristics.

The benchmark suite and analysis functions, as well as the reasons why these features have been selected as target characteristics, have to be stated. Before going ahead with this explanation, let us consider the general principles on which the solution is based.

First, classification accuracy of the acquired knowledge is used as the main basis for measurement. Second, the result of the learning process must be a body of knowledge that contains knowledge not directly explicit in the learning set. Third, the target characteristics should be important for the selected problems (the tasks specified in the benchmark suite). The first principle is used for analyzing results and the other two for choosing a good benchmark suite.

Benchmark Target Characteristics

The selection of the benchmark target characteristics depends mainly on the pursued goal and, in particular, on the user of the proposed model, that is, on the user of the benchmark. Assume that the knowledge engineer has been chosen as the user of the model. This assumption is, in fact, a realistic one because machine-learning tools can be of help to the knowledge engineer in knowledge-acquisition activities. However, if he/she wants to choose

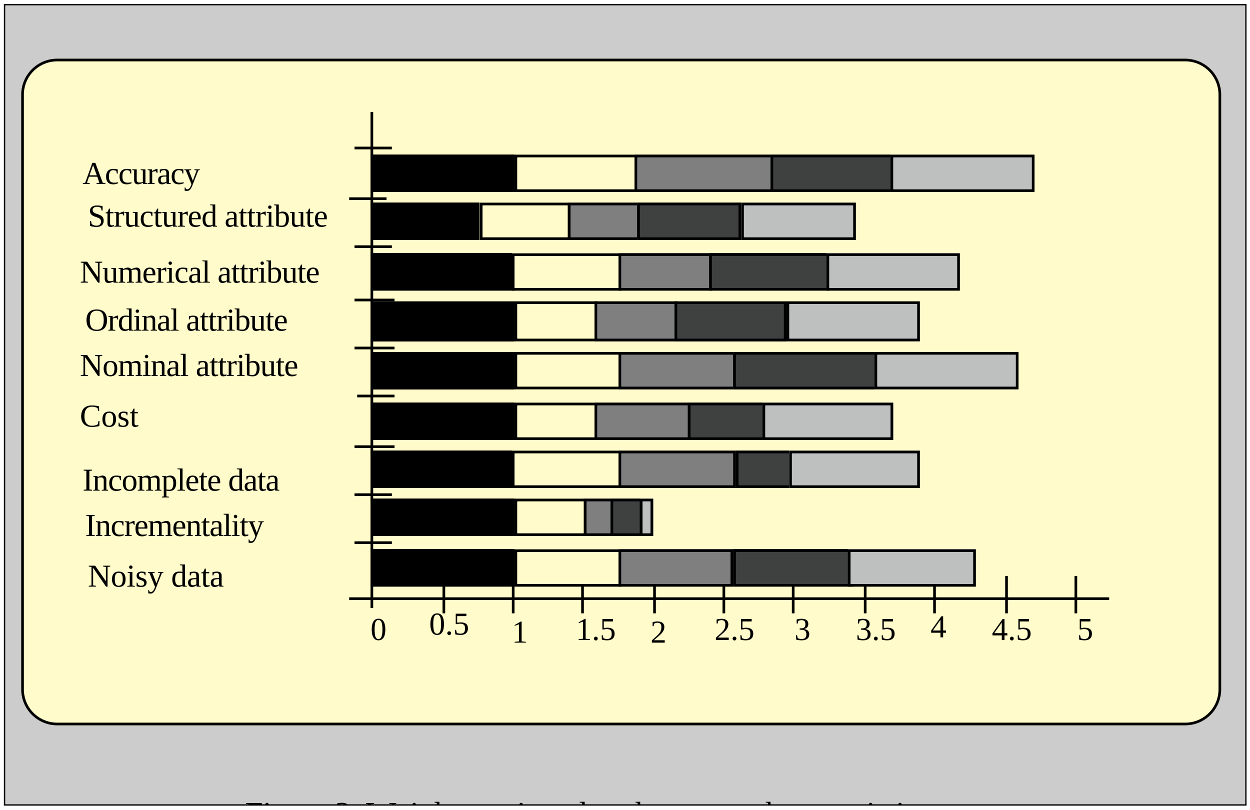

Figure 3. Weights Assigned to the Target Characteristics.

one tool from a set of them, what is he/she going to look for? The answer to this question constitutes the target characteristics.

Five experts were selected to help determine a good set of features. These experts are experienced knowledge engineers with a sound knowledge of machine learning and the use of these tools for knowledge acquisition. A list of possible features of interest was given to the experts, who assigned weights between zero and one to each of the characteristics; zero meant of no interest, and one meant really important. Figure 3 is a list of selected features and their respective weights.

There are other features that appear in the machine-learning literature but not in figure 3 (Muñoz 1991). This difference exists because the experts did not consider these characteristics relevant to the purpose at hand; that is, they assigned them a total weight lower than one.

First, *accuracy* refers to the classification accuracy of the acquired knowledge. A main goal in using these tools is to acquire precise knowledge; its importance is patent even for nonexperts.

Structured attributes are attributes whose values are not flat; that is, they form a structure (García 1991; Nuñez 1991, 1990; Witten and McDonald 1990). These structures usually appear as a hierarchy (Witten and MacDonald 1990). The presence of these structures is associated with the use of background knowledge (Nuñez 1991, 1990; Mellis 1989) to guide the generalizations. The grouping of villages into counties, counties into states, and so on, is a good example of these attributes.

Ordinal attributes (Cestnik 1987; Mingers 1989a) are attributes whose values can be ordered. A person's academic qualifications and how well he/she speaks a foreign language are examples of this kind of attribute. *Numeric attributes* can be considered as a kind of ordinal attribute in which the order is given by the greater-than relation in real numbers. Examples of these attributes (Cestnik 1989, 1987) are temperature, distance from a

		Number of classes	Number of attributes	Average number of values per attribute	Learning set size	Test set size
Ordinal attribute	Task 1	5	3	8	60	25
Numerical attribute	Task 1	3	2	60*	40	15
	Task 2	2	2	100*	169	30
Structured attribute	Task 1	2	4	5.25	12	10
	Task 2	3	3	8	50	30
Cost	Task 1	5	6	3	353	118**

* There is an infinity of possible values. This average corresponds to real numbers with only one decimal digit.

** There are three test sets, each with 118 test cases.

Table 2. Main Features of the Benchmark Suite.

point, and position on a plane.

The *nominal attributes* constitute the basis of machine-learning algorithms and consist of attributes whose values are symbols. There are no predefined relations between the values. Figure 3 shows the importance of these attributes.

The cost or economy factor (Nuñez 1990, 1988; Tan 1990; Tan and Schlimmer 1989) consists of the association of a cost to the test needed to establish an attribute value. For example, if someone needs to know the temperature of a place on earth and a place on the moon at different times to reach a conclusion, the cost of getting the temperature on earth will obviously be lower than the cost of getting it on the moon. A tool dealing with cost will solve a problem using the most economic information and will only consider higher-cost attributes when they are strictly necessary.

Noisy and incomplete data are related to the fact that the normal data source is the real world. These data (Mingers 1989b; Clark and Niblett 1987) usually come with some mistakes (noise) made during their collection and sometimes include unknown values for some of the collected features; that is, they are incomplete.

Finally, a machine-learning tool is incremental when the learning set does not need to be complete to start the learning process (Fernández 1990; Utgoff 1989, 1988). Thus, once the knowledge has been acquired with the initial learning set, it can be extended to cover other available examples without having to recompute all the examples contained in the initial set. As a result of this study, the benchmark has centered on the main features that a knowledge engineer wants to find in a machine-learning tool, leaving aside those that have already been dealt with in the literature to date. Thus, the tasks forming the benchmark suite differ from those appearing in the literature (López de Mántaras 1991; Shavlik, Mooney, and Towell 1991; Mingers 1989a, 1989b; Michalski et al. 1986), which are designed for other kinds of studies, for example, those on classification accuracy, noisy and incomplete data, or incremental learning.

Benchmark Suite

The benchmark suite specifies the tasks that must be carried out by the tools to arrive at results to which the analysis functions can be applied. As mentioned previously, each task consists of a learning set containing the initial examples that are given to the tools for

Ordinal attributes	$f(p) = \max(0, \frac{p - 0.2}{0.8})$
Numerical attributes	$f(x, y) = \begin{cases} 0 & x < 0.5 \\ \frac{(x + 2y)}{3} & x >= 0.5 \end{cases}$
Structured attributes	$f(x, y) = \sqrt{f_1(x) * f_2(y)}$ $f_1(x) = \max(\frac{x - 0.5}{0.5}, 0)$ $f_2(x) = \max(\frac{x - 0.33}{0.67}, 0)$
Cost	$f(c, p) = \sqrt[3]{g(c) * h(p)}$ $g(c) = \sqrt{1 - c^2}$ $h(p) = e^{(1.5 - (p + 1/2p^2))} + 0.05 \sin^2 (\Pi(1-p)^{1.8})$

Table 3. Analysis Functions.

learning purposes and a test set containing the examples used for test purposes. The benchmark suite specifies one or more tasks for each benchmark target characteristic. The main features of these tasks are summarized in table 2. The tasks have been selected carefully on the basis of the following:

First, the examples of both the learning and test suites will contain attributes for testing the associated target characteristics. Let us call this attribute A_s.

Second, the attributes A_s will be crucial for learning purposes during the selected task; that is, their information gain (the information obtained when their value is known) will be high.

Third, the examples contained in the learning and test suites will contain extreme conditions, which means that the learning set will contain just the minimum information needed for a human being to solve the task, and the test set will explore the examples that a human being finds most difficult to deal with.

Fourth, in view of these principles, the smaller the training and test cases are, the more suitable a task is for inclusion in the benchmark suite. Thus, if a simple task is found that fits the benchmark purposes perfectly, more complex ones are rejected.

Bearing in mind these guidelines, table 2 shows how some of the tasks are really simple in terms of the learning and test-set sizes, conveying a rich combination of attributes for the pursued goal. A detailed explanation of each task is beyond the scope of this article; however, a brief description, including the most relevant attributes for each task, follows.

Ordinal attributes are tested using one task that consists of assigning a job to a person depending on some of his/her abilities. Both the abilities considered (academic rank, job experience, and foreign languages) and the job to be assigned are ordinal.

For numeric attributes, there are two tasks, both involving numeric relations: (1) classifying figures on a surface depending on their position, size, and form and (2) classifying points on a surface previously divided into different areas. The first task is a simple one and merely aims to establish whether or not a tool deals with numeric attributes. The second one provides for a closer approximation, which will be taken into account by the analysis functions.

There are also two tasks for structured attributes. The first one is to classify a geometric figure based on its shape, color, size, and material. A hierarchical relation has been established among the values of the first three attributes, just as with the values of the solution. The second task consists of determining the skin color of a person, taking into account his/her place of birth, hairstyle, and eye color. All the attributes are structured; for example, if a person is born in Spain, he/she also belongs to the European community, to Europe, and so on; if he/she is born in Annapolis, he/she also belongs to Maryland, to the United States, to North America, and so on.

The cost or economy factor has an associated task consisting of determining whether a woman is pregnant according to a set of six attributes, all of them with an associated cost.

Results Analysis

Classification accuracy *p* was taken to measure the results with a view toward analysis. Once the knowledge has been acquired by the tools using the learning set, its classification accuracy *p* is determined using the test set. These values are then put into some specific functions to get meaningful measures. These functions are the analysis functions that are shown in table 3 and figures 4 to 7 for each target characteristic.

When the benchmark suite specifies two tasks, two parameters appear in the analysis function, and these parameters represent the accuracy obtained with the first and the second tasks, respectively. This value is standardized between zero and one. An exception is the function for cost analysis. Only one task is specified in the benchmark suite for this characteristic; however, the function has two arguments because in this case, each attribute has an associated cost, and the cost involved in the classification process is also an important feature to be considered. Table 3 shows all the analysis functions, and figures 4 to 7 sketch their graphic representations for those values on which the functions are defined.

Different functions have been tried to find the one that best determines the following aspects for each target characteristic: (1) when are the results good enough for a value of one to be given to the tested tool, (2) where is the line below which a value of zero is given to the tested tool, and (3) which is

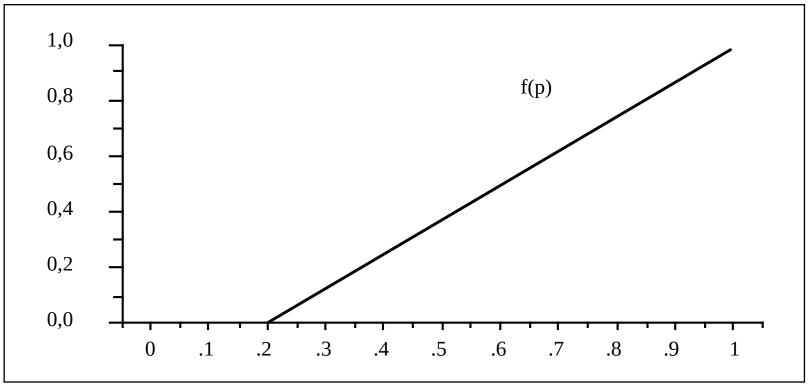

Figure 4. Graphic Representation of the Analysis Function for Ordinal Attributes.

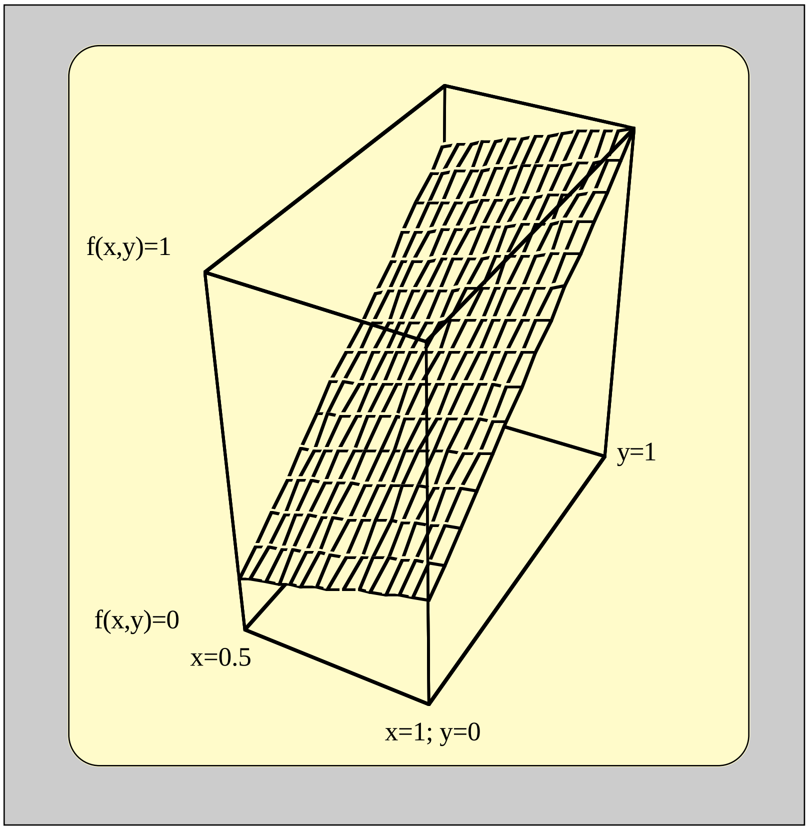

Figure 5. Analysis of Results for Numeric Attributes in the Interval in Which the Value of f(x,y) ≠ 0(x > 0.5).

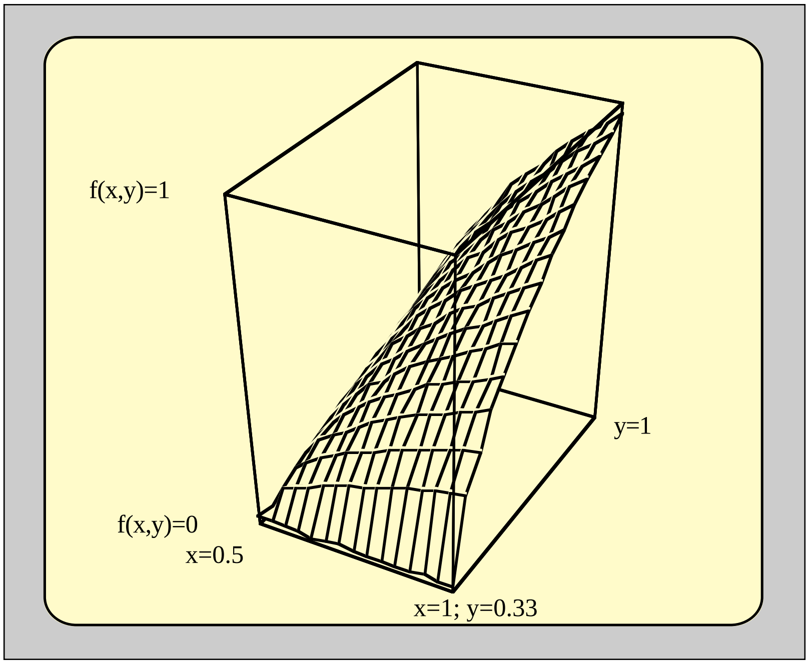

Figure 6. Analysis Function for Structured Attributes in the Interval in Which the Value of f(x,y) ≠ 0(x > 0.5; y > 0.33).

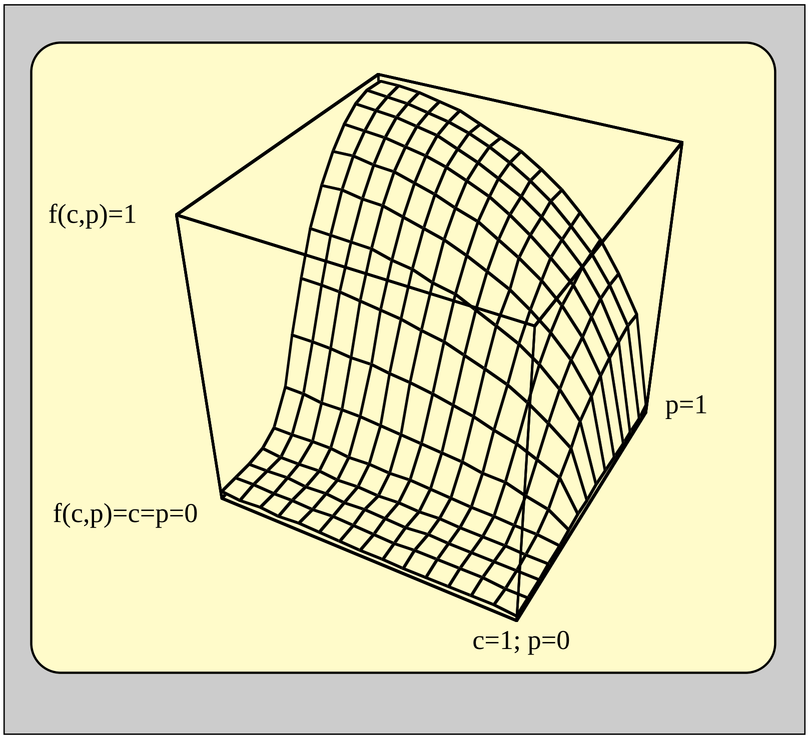

Figure 7. Analysis Function for Cost, *f(c,p)*.

the best transition path from zero to one.

For each target characteristic, all the functions shared the characteristic that their values rose at the same time as classification accuracy *p*. However, they differed in respect to the three previous aspects. Both experts and knowledge engineers played an important part in selecting the analysis functions shown in table 3, just as they did in experimentation.

Experiments and Results

Several experiments were carried out with a collection of machine-learning tools to validate the proposed benchmark. The experiments and results analysis are explained in the following paragraphs.

Selecting the Workbench

The first job was to select a set of tools for testing the benchmark. Two constraints were imposed on these tools: (1) there should be a reliable a priori reference on whether the benchmark target characteristics are present or not and (2) at least one of the selected tools must have these target characteristics. The aim behind these constraints was proper validation of the model. The a priori references were the tool developers themselves and the reference manuals. The first source is the most reliable one because the tool developer knows exactly what tool capabilities are. Fortunately, it was possible to contact the developers of three of the four tools used as a workbench.

Four tools were selected in this way: ASSISTANT 86 (Cestnik 1987), ALEXIS II (García 1991; Nuñez 1991), AQ15 (Álvaro 1990), and ID* (Fernández 1990). The set represents a collection containing the two mainstreams in machine learning from examples: the AQ family (AQ15) and the ID family (ASSISTANT 86, ALEXIS II, and ID*). The last tool groups three implementations of the ID family: ID3, ID4, and ID5. All these tools have been tested, but because there are no differences with respect to the benchmark, all three are now grouped under ID*. A full description of these tools is beyond the scope of this article; the main issue is to establish whether the benchmark target characteristics are present in these tools. Table 4 shows the a priori reference for these tools.

Experiment Results

Tables 5 and 6 show the results obtained with the benchmark on the selected tools workbench. Table 5 shows the values of the

	ASSISTANT 86	ALEXIS II	AQ15	ID*
Ordinal attribute	YES	YES	NO	NO
Numerical attribute	YES	YES	NO	NO
Structured attribute	NO	YES	NO	NO
Cost	NO	YES	YES	YES

Table 4. Presence of the Benchmark Target Characteristics: A Priori References.

		ASSISTANT 86	ALEXIS II	AQ15	ID*
Ordinal attribute	p	1	1	0.16	0
Numerical attribute	x	1,	0.8	0.33	0,
	y	0	0.9	-	-
Structured attribute	x	0,	1,	0.5	0,
	y	1	1	0.33	0
Cost	c	1,	0.72	0.85	0.51
	p	1	1	1	0.94

Table 5. Values of the Parameters Needed for the Analysis Functions.

		ASSISTANT 86	ALEXIS II	AQ15	ID*
Ordinal attribute	f(p)	1	1	0	0
Numerical attribute	f(x,y)	0.33	0.86	0	0
Structured attribute	f(x,y)	0	1	0	0
Cost	f(c,p)	0	0.88	0.81	0.95

Table 6. Analysis Function Values.

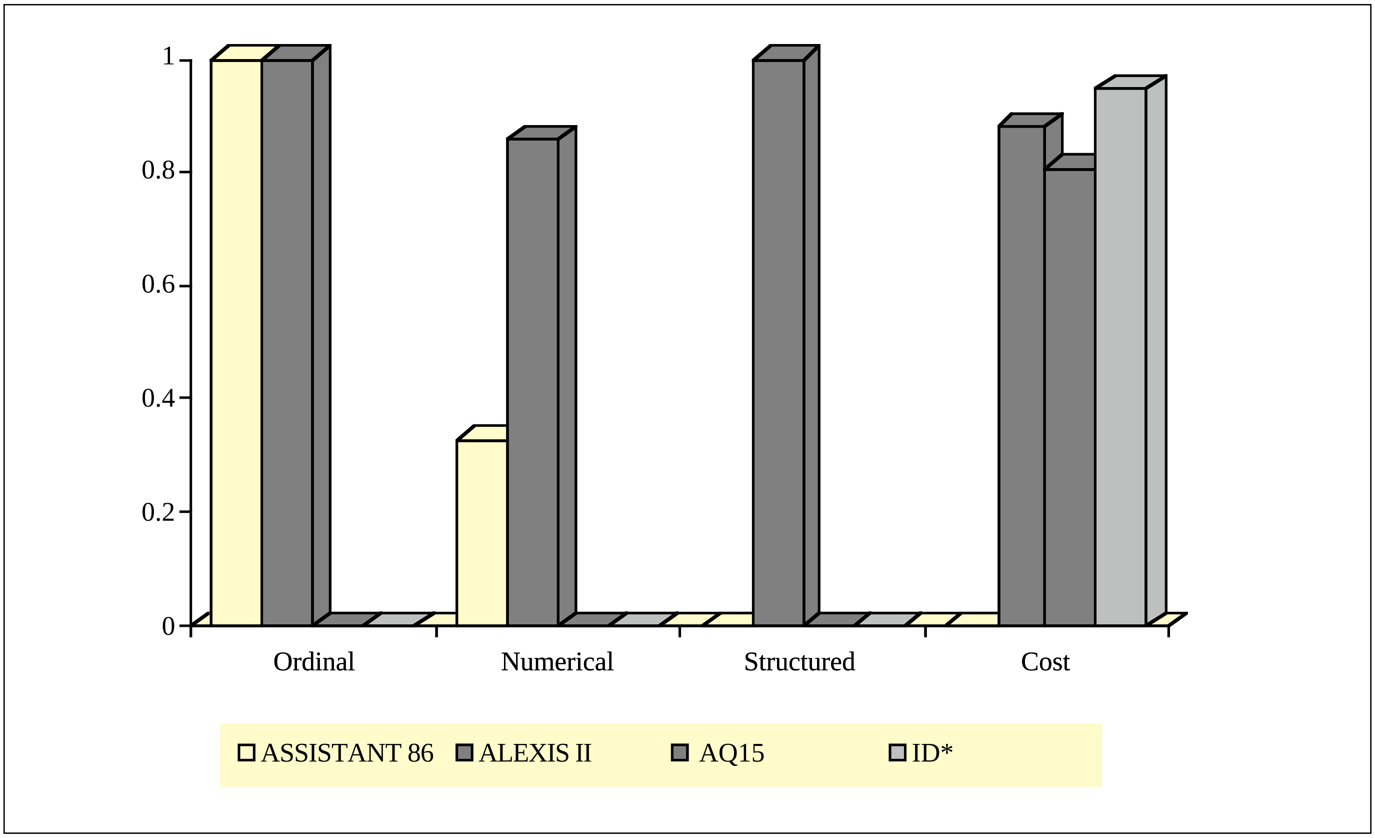

Figure 8. Comparative Strengths and Weaknesses of Tools.

parameters needed for the analysis functions, and table 6 shows the values of the functions themselves.

The - symbol in table 5 appears only in the numeric attribute row. Based on the analysis function for these attributes, if classification accuracy with the first task is lower than 0.5, the function value will automatically be zero, with no consideration of the classification accuracy of the second task. Thus, the second task does not need to be carried out for these two tools.

A comparison of tables 4, 5, and 6 illustrates the parallelism of tables 4 and 6, which suggests the adequacy of the model. Nevertheless, several points should be stressed. The first point is the value of ASSISTANT 86 in table 6 for numeric attributes: It is not as close to one as expected because an attribute can have any numeric value in ASSISTANT 86. However, when it is considered, the user must specify the interval bounds; that is, this information is not induced from the learning set. The benchmark suite and analysis function detect this fact and consider that it is not a good way of dealing with numbers. However, this value is greater than zero, which differentiates it from other tools that do not deal with numbers at all.

Table 5 shows that the values of *x* and *y* for ALEXIS II are too close, but these values for ASSISTANT 86 are different. The reason for this difference is that ALEXIS II deals with structured attributes in a reasonable way; ASSISTANT 86 does not deal with this type of attribute but provides some mechanisms that can simulate them under certain circumstances. The proposed benchmark is robust in this sense.

Table 5 shows how the values in the AQ15 column are not zero, but those in the ID* column are. However, in table 6, the value of the analysis function is zero for both of them. The differences between the two tools, as shown in table 5, are purely coincidental. The analysis function detects that they are chance differences because the values of table 6 are zero for both tools.

In the cost row of table 6, the best result is for ID* (.95), followed by ALEXIS II (.88). It is important to mention that according to their developers, both tools use the same cost function. Why is there this difference? It is because ID* deals with noisy and incomplete data, but ALEXIS II does not. This mechanism allows ID* to get more economic knowledge at the expense of slightly lower accuracy. A gain in economy is of greater interest than the loss of accuracy in this case; the analysis function gives better values for ID*. This result backs up previous studies (Mingers 1989a) that show how, in some circumstances, dealing with noisy data improves the results obtained.

Tuning the Benchmark

Even with good results, as explained previously, the value of the numeric-analysis function for ALEXIS II and the values of the cost-analysis function for ALEXIS II, AQ15, and ID* suggest that these functions can be tuned. For this purpose, the tool developers have been asked to quantify the a priori reference yes for these attributes. The result of this quantification for the numeric attributes has been YES = 1. The analysis function can be tuned by applying a factor a to it, such that a * 0.86 = 1; so, a = 1.16. With respect to cost, the tool developers accept the values obtained as good, except for ALEXIS II, whose value should be greater than or equal to .9. Thus, a small factor b can be applied to this function, such that b * 0.88 = 0.9; that is, b = 1.02. This factor also maintains the values for the other tools within the quantified range.

Figure 8 is a graph showing the comparative strengths and weaknesses of each tool included in the workbench. Based on the assumption that a knowledge engineer wants to choose one of these tools to solve a particular problem, this graph can be taken as a good basis for selection.

Conclusions

The benchmark presented constitutes an important step in the search for measurement criteria to be applied to any machine-learning tool. To be precise, it is the first benchmark applicable to machine-learning tools and provides an objective and reliable measure.

Taking the knowledge engineer as the user of the benchmark is a new approach to comparative studies, close to the focus of most current studies of the user perspective.

The experiments show that the model eliminates the random effect and detects the weakness of some tools that are supposed to deal with numeric attributes, such as ASSISTANT 86.

New evidence has been found to support the fact that in some circumstances, dealing with noisy data improves the results sought.

Finally, more cases can be added to the benchmark suite to allow comparative studies of characteristics other than ordinal, numeric, and structured attributes or cost considerations. This feature of the model is important.

Future Work

In view of the benchmark's extensibility, an important research effort would be to add new features to deal with other important characteristics, such as background knowledge and compound objects. We are, in fact, currently working along these lines. It would also be interesting to apply the benchmark to other tools that have been compared previously in other ways and to study both results.

... an important research effort would be to add new features to deal with other important characteristics, such as background knowledge and compound objects. We are, in fact, currently working along these lines.

Acknowledgments

This article was prepared and written in collaboration with the Centre of Technology Transfer in Knowledge Engineering in Madrid, Spain.

References

Alonso, F.; García, G.; Maté, J. L.; Morant, J. L.; and Pazos, J. 1990. Evaluation in Knowledge Engineering: Classifying, Comparative, and Metric Criteria. Presented at the Fourth International Symposium on Knowledge Engineering, 7–11 May, Barcelona, Spain.

Álvaro, R. 1990. Induction as a Solution for Knowledge Acquisition: AQ Algorithm. Master's thesis, Facultad de Informática, Universidad Politécnica de Madrid.

Atlas, L.; Cole, R.; Connor, J.; El-Sharkawi, M.; Mars, R. J.; Muthusamy, Y.; and Barnard, E. 1990. Performance Comparisons between Back-Propagation Networks and Classification Trees on Three Real-World Applications. *Advances in Neural Information Processing Systems* 2:622–629.

Bratko, I., and Lavrac, N. 1987. *Proceedings of EWSL '87: Second European Working Session on Learning.* Bled, Yugoslavia: Sigma.

Cendrowska, J. 1988. PRISM: An Algorithm for Inducing Modular Rules. *Knowledge-Based Systems* 1:225–276.

Cestnik, B. 1989. Informativity-Based Splitting of Numeric Attributes into Intervals. In *Expert Systems, Theory and Applications, IASTED 89*, ed. M. H. Hamza, 59–62. Anaheim, Calif.: Acta.

Cestnik, B. 1987. ASSISTANT PROFESSIONAL, A Software Tool for Inductive Learning of Decision Rules. System User Manual, Edvard Kardelj University, Ljubljana, Yugoslavia.

Clark, P., and Niblett, T. 1989. The CN2 Induction Algorithm. *Machine Learning* 3(4): 261–283.

Clark, P., and Niblett, T. 1987. Induction in Noisy

Domains. In *Proceedings of EWSL '87: Second European Working Session on Learning*, 11–30. Bled, Yugoslavia: Sigma.

Dhaliwal, J. S., and Benbasat, I. 1990. A Framework for the Comparative Evaluation of Knowledge-Acquisition Tools and Techniques. *Knowledge Acquisition* 2:141–166.

Dietterich, T. G., and Michalski, R. S. 1983. A Comparative Review of Selected Methods for Learning from Examples. In *Machine Learning: An AI Approach, Volume 1,* eds. R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, 41–81. San Mateo, Calif.: Morgan Kaufmann.

Fernández, A. 1990. Comparative Study of Prune Methods for Decision Tree Induction Algorithms. Master's thesis, ETSI Telecomunicaciones, Universidad Politécnica de Madrid.

Fisher, D. H., and McKusik, K. B. 1989. An Empirical Comparison of ID3 and Back Propagation. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 788–793. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Gams, M., and Lavrac, L. 1987. Review of Five Empirical Learning Systems within Proposed Schemata. In *Proceedings of EWSL '87: Second European Working Session on Learning,* 46–78. Bled, Yugoslavia: Sigma.

García, L. E. 1991. A Modification to the ID3 Algorithm for Decision Tree Induction. Master's thesis, ETS Ingenieros de Telecomunicación, Universidad Politécnica de Madrid.

Hayes-Roth, F. 1989. Towards Benchmarks for Knowledge Systems and Their Implication for Data Engineering. IEEE *Transactions on Knowledge and Data Engineering* 1(1): 101–110.

Kodratoff, Y., and Michalski, R. S. 1990. *Machine Learning: An Artificial Intelligence Approach, Volume 3*. San Mateo, Calif.: Morgan Kaufmann.

López de Mántaras, R. 1991. A Distance-Based Attribute Selection Measure for Decision Tree Induction. *Machine Learning* 6(1): 81–92.

Mellis, W. 1989. A General Approach to the Use of Background Knowledge in a Numerical Induction Algorithm. Presented at the Third European Workshop on Knowledge Acquisition for Knowledge Based Systems, EKAW-90, July, Paris, France.

Michalski, R. S. 1990. Learning Flexible Concepts: Fundamental Ideas and a Method Based on Two-Tiered Representation. In *Machine Learning: An Artificial Intelligence Approach, Volume 3,* eds. Y. Kodratoff and R. S. Michalski, 63–111. San Mateo, Calif.: Morgan Kaufmann.

Michalski, R. S.; Carbonell, J. G.; and Mitchell, T. M. 1983. *Machine Learning: An Artificial Intelligence Approach, Volume 1*. San Mateo, Calif.: Morgan Kaufmann.

Michalski, R. S.; Mozetic, I.; Hong, J.; and Lavrac, N. 1986. The Multipurpose Incremental Learning System AQ15 and Its Testing Application to Three Medical Domains. In Proceedings of the Fifth National Conference on Artificial Intelligence, 1041–1045. Menlo Park, Calif.: American Association for Artificial Intelligence.

Mingers, J. 1989a. An Empirical Comparison of Selection Measures for Decision Tree Induction. *Machine Learning* 3: 319–342.

Mingers, J. 1989b. An Empirical Comparison of Pruning Methods for Decision Tree Induction. *Machine Learning* 4: 227–243.

Mooney, R. J.; Shavlik, J. W.; Towell, G. G.; and Grove, A. 1989. An Experimental Comparison of Symbolic and Connectionist Learning Algorithms. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 775–780. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Muñoz, P. L. 1991. Elaboration of a Benchmark for Machine-Learning Tools. Ph.D. diss., Facultad de Informática, Universidad de Politécnica de Madrid.

Nuñez, M. 1991. The Use of Background Knowledge in Decision Tree Induction. *Machine Learnin*g 6(3): 231–250.

Nuñez, M. 1990. Decision Tree Induction Using Domain Knowledge. In *Current Trends in Knowledge Acquisition*, eds. B. Wielinga, J. Boose, B. Gaines, G. Schreiber, and M. van Someren, 276–288. Amsterdam: IOS.

Nuñez, M. 1988. Economic Induction: A Case Study. In *Proceedings of EWSL '88: Third European Working Session on Learning*, 139–145. San Mateo, Calif.: Morgan Kaufmann.

O'Rorke, P. 1982. A Comparative Study of Inductive Learning Systems AQ11P and ID3 Using a Chess Endgame Test Problem, UIUCDCS-F-82899, Department of Computer Science, University of Illinois at Urbana-Champaign.

Quinlan, J. R. 1989. Unknown Attribute Values in Induction. In *Proceedings of the Sixth International Workshop on Machine Learning,* 164–168. San Mateo, Calif.: Morgan Kaufmann.

Quinlan, J. R. 1986. The Effect of Noise in Concept Learning. In *Machine Learning: An AI Approach, Volume 2,* eds. R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, 149–169. San Mateo, Calif.: Morgan Kaufmann.

Shavlik, J. W.; Mooney, R. J.; and Towell, G. G. 1991. Symbolic and Neural Learning Algorithms: An Experimental Comparison. *Machine Learning* 6:111–143.

Tan, M. 1990. CSL: A Cost-Sensitive Learning System for Sensing and Grasping Objects. Presented at IEEE International Conference on Robotics and Automation, 13–18 May, Cincinnati, Ohio.

Tan, M., and Schlimmer, J. C. 1989. Cost-Sensitive Concept Learning of Sensor Use in Approach Recognition. In *Proceedings of the Sixth International Workshop on Machine Learning,* 392–395. San Mateo, Calif.: Morgan Kaufmann.

Utgoff, P. E. 1989. Incremental Induction of Decision Trees. *Machine Learning* 4:161–186.

Utgoff, P. E. 1988. ID5: An Incremental ID3. In *Proceedings of the Fifth International Conference on Machine Learning,* 107–120. San Mateo, Calif.: Morgan Kaufmann.

Wielinga, B.; Boose, J.; Gaines, B.; Schreiber, G.; and van Someren, M. 1990. *Current Trends in Knowledge Acquisition.* Amsterdam: IOS.

Witten, I. H., and McDonald, B. A. 1990. Using Concept Learning for Knowledge Acquisition. In *Machine Learning and Uncertain Reasoning: Knowledge Based Systems, Volume 3,* eds. B. Gaines and J. Boose, 139–164. San Diego, Calif.: Academic.

Fernando Alonso Amo is professor of computer science and AI at the Universidad Politécnica de Madrid (UPM), where he also received his Ph.D. He is currently research and development director at UPM's Centre of Technology Transfer in Knowledge Engineering. He previously held several management posts at the Spanish Ministry of Education and Science Data Processing Centre. Author of several books on programming methodology and papers on software and knowledge engineering, his research interests lie in the application of AI techniques and benchmarks to improving the quality of life for the disabled, especially the blind.

Natalia Juristo is associate professor of computer science and AI for the Universidad Politécnica de Madrid (UPM) Faculty of Computer Science. She also received her Ph.D. from UPM. She gained her professional experience at the Centre d'Études pour la Recherche Nuclear CERN in Geneva (1988); the European Space Agency (ESA) in 1989; and the Carnegie Mellon University Software Engineering Institute in 1992, where she was a visiting lecturer. Coauthor of a book on AI and another on knowledge engineering, her research interests lie in knowledge acquisition, software system evaluation in general, and expert systems. An advocate for cooperation between knowledge engineering and software engineering, she is academic director of the Master's Program in software engineering and knowledge engineering that has been conducted for eight years at UPM.

Luis Maté is professor of computer science and AI at the Universidad Politécnica de Madrid (UPM) and is, at present, dean of the Faculty of Computer Science. He received his Ph.D. from UPM and founded its Computer Centre. His work on information system design is one of his most prominent activities, an area on which he advises several Spanish and international institutions, including the upper and lower houses of the Spanish Parliament. Author of several books, he is an advocate of the integration of software and knowledge engineering.

Pedro Luis Muñoz is a researcher at Telefónica Investigación y Desarrollo. He received his Ph.D. from the Universidad Politécnica de Madrid in 1991. He has been employed as an assistant professor of AI, expert systems, and functional programming and has worked in the Artificial Intelligence Laboratory at Madrid's Faculty of Computer Science and in the Cybernetics Research Laboratory at the University of Maryland at College Park. His research interests include knowledge acquisition, evaluation, expert systems, and network management.

Juan Pazos is professor of computer science and AI at the Universidad Politécnica de Madrid (UPM) and director of AI research. He received his Ph.D from UPM. A visiting lecturer at over 20 universities and research centres, including Carnegie Mellon, Sunderland, and IRIA, his research interests lie in heuristic search; knowledge engineering; and, in particular, expert system building methodology and evaluation and the theoretical foundations of AI. He is coauthor of two books on AI and another on knowledge engineering.