VI INDUCTION AND REASONING FROM CASES

Michel MANAGO [1], Klaus-Dieter ALTHOFF [2], Eric AURIOL [1], Ralph TRAPHÖNER [3], Stefan WESS [2], Noël CONRUYT [1], Frank MAURER [2]

1 Introduction

We present the INRECA european project (ESPRIT 6322) on integration of induction and case-based reasoning (CBR) technologies for solving diagnostic tasks. A key distinction between case-based reasoning and induction is given in [1]: "In case-based methods, a new problem is solved by recognising its similarities to a specific known problem then transferring the solution of the known problem to new one (...) In contrast, other methods of problem solving derive a solution either from a general characterisation of a group of problems or by search through a still more general body of knowledge". In this paper, we distinguish between a pure inductive approach and a case-based one on the basis that induction first computes an abstraction of the case database (ex: a decision tree or a set of rules) and then uses this general knowledge for problem solving. During the problem solving stage, the system does not access the cases.

2 INRECA's inductive and case-based approaches

Induction is a technology that automatically extracts general knowledge from training cases. KATE is the inductive component of INRECA. It builds a decision tree from the cases by using the same search strategy, hill-climbing, and same preference criteria that is based on Shannon's entropy as ID_3 [2]. Unlike most induction algorithms, KATE can handle complex domains where cases are represented as structured objects with relations and it can use background knowledge. At each node, KATE generates the set of relevant attributes of objects for the current context and selects the one that yields the highest information gain. For instance, an attributes such as “pregnant” for a patient whose sex is known to be “male” further up in the decision tree is eliminated before the information gain computation. Background domain knowledge and class descriptions allow to constrain the search space during induction [3].

Case-based reasoning is a technology that makes direct use of past experiences to solve a new problem by recognising its similarity with a specific known problem and by applying the known solution to the new problem. PATDEX is the case-based component of INRECA. It consists of two case-based reasoning subcomponents for classification and test selection. A procedure that dynamically partitions the case base enables an efficient computation and updating of the similarity measures used by the CBR subcomponents. For the classification subcomponent, the applied similarity measures are dynamic. The underlying evaluation

[1] AcknoSoft , 58a rue du Dessous des Berges, 75013 Paris - France. [2] University of Kaiserslautern, dept. of Computer Science, PO Box 3049, 6750 Kaiserslautern - Germany. [3] tecInno GmbH, Sauerwiesen 2, 67661 Kaiserslautern - Germany.

function is adapted using a connectionist learning technique (competitive learning). For the test selection, the adaptation of similarity measures is based on an estimation of the average costs for ascertaining symptoms using an A*-like procedure. PATDEX can deal with redundant, incomplete, and incorrect cases and includes the processing of uncertain knowledge through default values. PATDEX is described in [4] and [5].

3 The need for integration

INRECA integrates induction and case-based reasoning so that they can collaborate and provide better solutions than they would individually. Before describing how integration is performed, we first state why the two approaches are complementary. Induction presents some limitations for building an identification system that can handle missing values during consultation. Consider the following case base drawn from an application that identifies marine sponges developed at the Museum of Natural History in Paris.

CASE	CLASS	SHAPE(BODY)	TEETH-TIP(MACRAMPHIDISQUES)	...
Ex1PARADI	SCONEMAELLIPSOID	LARGE	…	
Ex2COSCIN	ONEMACONICAL	LANCET-SHAPE	…	
Ex3CORYN	ONEMAELLIPSOID	LANCET-SHAPE	…	
… …	…	…	…	

Table 1 - A database of cases for an application which identifies marine sponges

KATE works in two steps: it first learns a decision tree and then uses the tree to identify the unknown class of a new incoming sponge. Consider what happens when the user does not know how to answer the first question asked during consultation of the tree of figure 1.

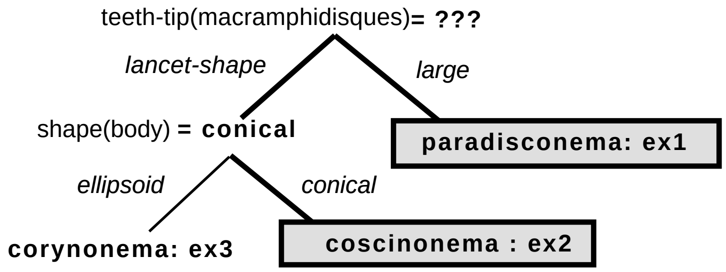

Figure 1: A consultation of the decision tree learned by KATE

When the user answers "unknown", KATE proceeds by following both branches "lancet-shape" and "large" and combines the conclusions found at the leaves. In the "large" branch, it reaches the "Paradisconema" leaf node. In the "lancet-shape" branch, it reaches a test node and the user is queried for the value of the "shape" of the object "body". He answers "conical". KATE reaches the "Coscinonema" leaf and combines the two leaves to conclude that the current case is a "Paradisconema" with a probability of 0.5 or a "Coscinonema" with a probability of 0.5. Consider case ex1 at the "Paradisconema" leaf node. The feature "shape(body)" of ex1 has the value "ellipsoid" unlike the current case where it is "conical". Thus, the current case is closer to ex2 than to ex1 and the correct conclusion is "Coscinonema" with a probability of 1. Unfortunately, the information about the “body shape” of ex1 was generalized away during induction and is no longer available during consultation.

Note that there are other methods for handling unknown values during consultation of a tree. Instead of combining branches, one can assign a probability to the branches [6] and follow the most probable one. However, this does not remove the problem presented above. This problem is not caused by a flaw of the particular induction algorithm used by KATE since we could have used another algorithm and encounter a similar problem. It is not a flaw of the decision tree representation formalism since we could have used production rules generated automatically or manually and still run into this same problem. It is caused by the fact that we are reasoning using an abstraction of the training cases and have generalized away and thus lost some discriminant information. If the consultation system is to handle any configuration of unknown values, such as for applications that deal with photo-interpretation of objects whose features may be hidden in any combinations, case-based reasoning will always perform better than rule-based, decision tree-based or even neural network-based identification systems.

This has been confirmed by a set of experiments conducted using PATDEX. We have measured its ability to reach a correct solution when the working case is incomplete (i.e. contains unknown values). Experiments have been conducted with a training set of one hundred cases. The test set also consists of one hundred cases. For every test case the number of known symptom values has been stepwise reduced. Classification accuracy is measured against reduction of the presented information. The results are shown in table 1. Here, a reduced information of 70% means that every case is classified based on 30% of its known symptom values (where 60% of such cases have been correctly classified).

Reduced information (%)	0	10	20	30	40	50	60	70	80	90	100
Classification accuracy (%)	100	99	97	96	91	90	76	60	28	11	0

Table 2 - Measuring Correctness against Reduction of Information

As confirmed by this set of experiments, up to a certain limit, classification accuracy is not significantly decreased by reducing the number of known attribute values in the current case. For instance, when half of the values are missing the system still correctly identifies 90% of the test cases. When using induction, a single missing value for an attribute in the decision tree (this corresponds to a 0.5% reduction in the information available) yields a loss of 50% in accuracy. When a feature is unknown, a case-based reasoning tool looks for alternative features to identify the current case. CBR reacts dynamically and exploit all the information available. In addition, a CBR system is more resilient to errors made by the user during consultation since it computes a similarity measure from the global description of the cases and not a minimal subset like with the inductive approach. It can confirm the conclusions by asking additional questions that modify the similarity measure accordingly.

This does not imply that CBR always performs better than induction. During the first year of INRECA, we have defined a catalog of industrial criteria to conduct experiments and compare the two technologies. Our criteria catalog does not merely adresses technical issues such as performance and effectiveness, but also ergonomic and economic aspects such as user acceptance of the technology (domain specialist, naive end-user, data clerk, case engineer etc.), ease to build, validate and maintain the application and so on. After analysis, we claim that induction and CBR are complementary techniques and that integrating these will improve their standalone capabilities. Our comparison is summarized in the next section. The criterias have been introduced in hierarchical weighted grids to compare in an objective and exhautive manner the induction and CBR components of INRECA as well as other existing tools.

4 Comparison of induction and CBR

We summarize the respective merits of the techniques in the following table. Although the experiments have been conducted using PATDEX and KATE, the conclusions drawn are applicable to the underlying technologies in general. Note that according to the distinction between induction and CBR that has been explained in the introduction, we view tools that access the training cases to incrementally maintain the induced rules or trees as CBR tools.

Advantages of PATDEX (CBR)	Advantages of KATE (Induction)
The application is always up-to-date because CBR can work incrementally.	The consultation is consistent: what is true today will be true tomorrow (unless the tree has been updated).
CBR handles missing values during consultation and makes optimal use of the information available.	The decision tree can be compiled into a runtime that does not require the case base to do diagnosis. It can be easily integrated in the customer's environment.
CBR can widen the set of current hypothesis whereas induction only shrinks it.	The system supports exploratory data analysis and does consistency checks in the data base.
The CBR consultation is more flexible for the user of the consultation system. It can be driven by the user who supply the information he wants instead of being guided step by step through a decision tree. It can handle sensor input and react dynamically to the data.	The domain specialist can influence or even impose how the consultation is done by modifying the tree by hand. He controls the consultation process.
The CBR consultation is more resilient to errors. After finding a conclusion, the current solutions can be confirmed or refuted.	A classification of the data can be constructed based on the information contained in the tree.
Analogies can be made based on the whole case description instead of a minimal subset.	Induction produces a generalisation of the cases and turns data into knowledge.
The similarity measure used by PATDEX can evolve over time and is adaptable.	
The current consultation can be explained to the user by presenting previous cases.	The current consultation can be explained to the user by presenting the classification rule.
CBR interprets cases dynamically.	The consultation of the learnt tree is more performant than the CBR consultation

Table 3 - Cost-Benefit Analysis of Induction and CBR

5. Integrating induction and CBR

Four critical levels of integration have been identified. For the first level, the two techniques are seating side-by-side and are provided as stand-alone modules that work on the same case data expressed in the CASUEL object-oriented language (**toolbox strategy**). This is useful because a single technique may match the user’s needs for a particular application, while a combination of both may not. In addition, a decision tree produced by induction allows to detect the inconsistencies of a case database before its use by a case-based reasoning module. For the second level of integration, the two techniques are able to exchange results via the CASUEL representation language (**cooperative strategy**). The results of one may help to improve the efficiency and to extend the classification capabilities of the other. More precisely, a decision tree produced by induction can speed up the consultation by the case-based reasoner. The case-based reasoner can supplement the decision tree when choosing among different conclusions (case-based reasoning is started at the end of the consultation of the tree or during consultation when encountering unknown values). The third level of integration allows the combination of individual modules of the tools (**workbench strategy**). For instance, the information gain measure module may be used to choose the next attribute to be asked during an interactive CBR consultation. The last level fulfils the final goal of INRECA (**seamless integration**) by mixing the most relevant parts of the two technologies in a single system. Two critical modules are identified: the information gain computation module for the induction technique, and the similarity computation module for the case-based reasoning technique.

Our main point is that a single system will never meet the needs of everyone. INRECA offers several integration possibilities and must be configured to meet the requirements of a particular application or of a particular category of users. For instance, a naive end-user must be guided step-by-step by the consultation system in a decision-tree like fashion. On the other end, a domain specialist wants to directly supply whatever information he feels is relevant and remain in control of the consultation system. Moreover, what may be viewed as an advantage of a technology in a given context may turn out to be a drawback in another. For instance, incrementality can be seen as an advantage of CBR over induction to maintain the consulation system automatically and keep up with the knowledge that workers learn through their daily experience. On the other end, we are currently working with an equipement manufacturer who distributes the diagnostic system to his customers and who wants to control the advices that are given to the users (let it be for legual reasons). Thus, he prefers a system that does not evolve permanently and that behaves in a predictable way. In that context, the incrementality is a drawback since he wants to compile the case data into an induction tree that is maintained by him periodically. Finally, one technique may be better adapted at a specific stage of the application life cycle (for example, CBR at the begining to enrich the case database) but not at a later stage (for example, induction can compile the case database when it becomes too big and when efficiency becomes a problem). Thus, INRECA provides several options for the four levels of integration and can be configurated by the application developper . In the next section, we present an architecture that deals with the problem of handling unknown values using CBR, but that pre-index the cases using a decision tree for efficiency.

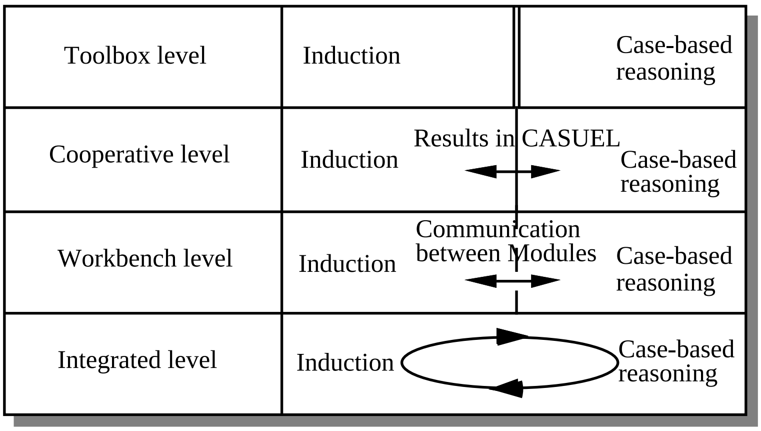

Figure 2. Four integration levels between Kate and Patdex

6. An integration architecture to handle missing values efficiently

As stated in section 3, one main drawback of a decision tree consultation occurs if the user answers “unknown” to a test. Unknown values propagate an uncertainty along all the branches of the “unknown node" - we define an unknown node as a node where the user answers “unknown” during the consultation of the tree although a subsequent test may remove this uncertainty. Moreover, the final diagnosis is probabilistic which is confusing for a non expert user. One way to deal with unknown values in the consultation of a tree is to switch to a case-based reasoning procedure after consulting the tree. When an unknown value is encountered, the consultation of the tree is stopped and the case-based reasoner is used to choose the next tests. The probabilistic diagnoses delivered by Kate may also be refined by using the similarity measure of the case-based reasoner. A workbench integration is needed. The procedure when encountering an unknown value in the consultation of the decision tree is presented below:

```
1.  Get the current situation given by the first tests
    of the tree.
2.  Get the current subset of the cases listed under the
    unknown node.
3.  Switch to Patdex by using the current situation and
    the current set of cases.
```

Procedure for Switching between Kate and Patdex

This procedure combines the advantages of both techniques for efficiency and correctness. In the worst case, the user answers unknown at the root node and we are left with a classical CBR consultation. In the best case, the user never answers unknown and we are left with a classical decision tree traversal mechanism that is very efficient.

Conclusions

Induction and case-based reasoning are complementary approaches for developing experience-based diagnostic systems. Induction *compiles* past experiences into general knowledge used to solve problems. Case-based reasoning directly *interprets* past experiences. Both technologies complement each other. Induction is used for detecting inconsistencies in the case data base, case-based reasoning is used during consulation to retrieve similar cases when there are missing values. The induction system can compute a tree to index cases on a predefined number of levels in order to improve the efficiency of case-based reasoning. After traversing that partial tree (interactive consultation), we are left at a leaf node with an initial candidate set that can be passed to the case-based reasoning system. As a consequence, the case-based reasoner works on a much smaller set of candidates. The partial decisions can be confirmed or refuted by the case-based reasoner. In the latter case the tree needs to be updated.

Acknowledgement

Funding for INRECA has been provided by the Commission of the European Communities (ESPRIT contract P6322). The partners of INRECA are AcknoSoft (prime contractor, France), tecInno (Germany), Irish Medical Systems (Ireland), the University of Kaiserslautern (Germany). KATE is a trademark of Michel Manago. We thank Prof. Claude Lévi and Mr Jacques Le Renard at the Museum of Natural History in Paris for providing the sample application used to illustrate some of the ideas presented here. We also thank Mr Thomas Schultz who has helped us refine our criteria list and who validated and filled our comparison grids for several CBR tools.

References

[1] Bareiss, R. (1989). Exemplar-Based Knowledge Acquisition. London: Academic Press

[2] Quinlan, R. (1983) Learning efficient classification procedures and their application to chess end games. In R. S. Michalski, J. G. Carbonell & T. M. Mitchell (Eds), *Machine Learning: An Artificial Intelligence Approach* (Vol. 1). Morgan Kaufmann.

[3] Manago M. (1989). "Knowledge Intensive Induction", proceedings of the sixth "International Machine Learning workshop", Morgan Kaufmann.

[4] Althoff, K.-D. & Wess, S. (1991). “Case-Based Knowledge Acquisition, Learning and Problem Solving in Diagnostic Real World Tasks”. *Proc. EKAW-91, Glasgow & Crieff*; also: GMD-Studien Nr. 211 (edited by M. Linster and B. Gaines)

[5] Richter, M. M. & Wess, S. (1991). “Similarity, Uncertainty and Case-Based Reasoning in PATDEX”. *Automated Reasoning - Essays in Honor of Woody Bledsoe*, Kluwer Academic Publishers

[6] Quinlan, J. R. (1989). “Unknown Attribute Values in Induction”. *Proceedings. of the Sixth International Workshop on Machine Learning,* pp. 164-168,. Morgan-Kaufmann.