Integrating Rules and Cases for the Classification Task

Jerzy Surma
Department of Computer Science
University of Economics
Komandorska 118/120, Wroclaw, Poland
Email: surma@ksk-2.iie.ae.wroc.pl

Koen Vanhoof
Faculty of Applied Economics Science
Limburgs University Center
B-3590 Diepenbeek, Belgium
Email: vanhoof@rsftew.luc.ac.be

Abstract. The recent progress in Case- Based Reasoning has shown that one of the most important challenges in developing future AI methods will be to combine and synergistically utilize general and case-based knowledge. In this paper a very rudimentary kind of integration for the classification task, based on simple heuristics, is sketched: "To solve a problem, first try to use the conventional rule-based approach. If it does not work, try to remember a similar problem you have solved in the past and adapt the old solution to the new situation". This heuristic approach is based on the knowledge base that consists of rule base and exception case base. The method of generating this kind of knowledge base from a set of examples is described. The proposed approach is tested, and compared with alternative approaches. The experimental results show that the presented integration method can lead to an improvement in accuracy and comprehensibility.

1 Introduction

The recent progress in Case- Based Reasoning has shown that one of the most important challenges in developing future AI methods will be to combine and synergistically utilize general and case- based knowledge (Aamodt 1995). The presented approach has an origin in Riesbeck and Schank's psychological consideration (Riesbeck & Schank 1989, pp.11): " When an activity has been repeated often enough it becomes rule- like in nature. We do not reason from prior cases when well- establish rules are available. [...] When the rule fails, the only alternative for its user is to create a case that captures that failure". Let us assume that our task is classification, and the knowledge is represented in: *rules* - that represent a standard and/or a typical situation, and *cases* - that represent the particular experience, exceptions and/or non-typical situations. The problem solver can classify a new case by means of the following algorithm:

If a new case is covered by some rule
Then apply a solution from a rule with the highest priority
Else adapt the solution from the most similar case

This algorithm is based on the following heuristics: "To solve a problem, first try to use the conventional rule- based approach. If it does not work, try to remember a similar problem you have solved in the past and adapt the old solution to the new

situation". The rules are evaluated first because standard situations occur more often, so it is more probable that input case is a standard case.
In section 2., we present an overview of related work on integrating Case- Based and Rule- Based Reasoning. In section 3 the different approaches for splitting Case Base are introduced. In section 4 we show the results of comparisons between the integrating approach and alternative approaches. The paper concludes with a short presentation of an algorithm learning from failures, and a discussion about the advantages and disadvantages of the suggested approach.

2 Related Work

The mixed paradigm involving case- based and traditional rule- based reasoning was included in the original CBR systems. The CHEF contained a rule- based sub module to support Case- Based Reasoning (Hammond 1988), and the CASEY used cases to supplement rule- based mechanism (Koton 1988).
The advanced studies in this field were made in the 1990s. Rissland and Skalak described a system CABARET that integrates reasoning with rules and reasoning with previous cases (Rissland & Skalak 1991). This integration was performed via a collection of control heuristics. Golding and Rosenblum propose the architecture for combining Rule- Based and Case- Based Reasoning for the task of pronouncing surnames (Golding & Rosenblaum 1991). The central idea of their approach is to apply the rules to a target problem to get a first approximation to the answer, but if the problem is judged to be compellingly similar to a known exception of the rules, then the solution is based on the exception rather than the rules. A good example of combined reasoning in the framework for problem solving in knowledge rich environment is the CREEK system (Aamodt 1991). Here, the system first attempts to solve the problem by case- based reasoning, and if an acceptable match is not found, rule based reasoning is attempted.
The selected studies in inductive learning were based on an integration with the case-based approach. Utgoff showed how the updating costs of incremental decision tree algorithms can be significantly decreased by saving specific instances (Utgoff 1989). Cardie proposed a method for using decision trees to specify the features to be included in k- nearest neighbor retrieval (Cardie 1993). The classic application of the decision tree as an index for case- based retrieval is implemented in commercial CBR tool ReMind, described and evaluated by Barletta (Barletta 1994).
In Europe a numbers of systems has recently been constructed, exploring various approaches of case and rule integration, including Cabata (Lenz 1993), and BUBE/CcC+ (Bamberger & Goos 1993). Malek and Rialle developed a computer-aided medical diagnosis system for neuropathy diseases where the domain knowledge is represented in prototype cases, and non- typical cases. During a diagnosis phase prototypes matched to the presented case are extracted, if no prototype is matched then the group of non- typical cases is retrieved (Malek & Rialle 1994). Armengol and Plaza presented the general knowledge modelling framework for solving the purification task of proteins. This task can be solved in different ways, especially by combining case- based retrieval and by using domain knowledge in the form of prototypes that are generated by an inductive method (Armengol & Plaza 1994).

The four possible levels of integration between the Induction and Case- Based Reasoning method were established and tested in the INRECA project (Manago et al., 1993), (Auriol et al., 1994). In terms of stand- alone, co-operative, workbench and seamless levels of integration, the approach presented in this paper is co-operative. This means that both methods are kept separated but they co-operate.
Some of the studies in machine learning that are concerned with the instance prototypicality are very important in the context of this research. Especially Matwin's and Plante's prototypical and marginal examples (Matwin & Plante 1994), Zhang's measure of instance typicality (Zhang 1992), and more recently Biberman's prototypicality approaches (Biberman 1995).

3 Integrating Rules and Cases

3.1 The Main Idea

The main source of knowledge for the problem solving heuristic (introduced in section 1) is a set of rules (Rule_Base) and a set of exception cases (Exception_Case_Base). In the traditional approach the rule base is obtained after a difficult and time consuming knowledge acquisition process. The main idea of the integrated approach is shown in Fig.1.

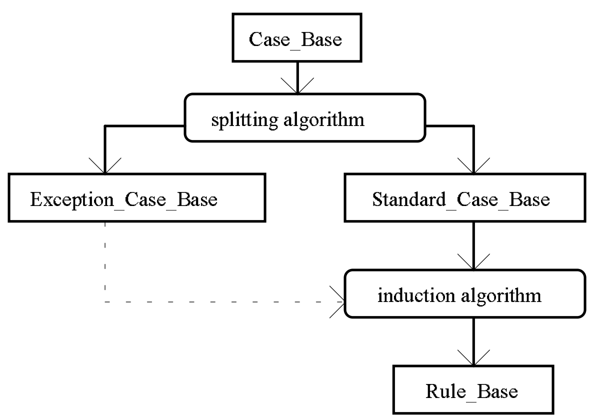

Fig.1. Generating Exception_Case_Base and Rule_Base

The given set of cases (Case_Base) is split into two disjoined subsets: Exception_Case_Base and Standard_Case_Base, and then the induction algorithm (e.g. C4.5) generates rules (Rule_Base) from standard cases (Standard_Case_Base). Now we can describe the problem solving heuristics from section 1 more precisely. To classify a new case, the ordered list of rules from the Rule_Base is examined to find the first whose condition is satisfied by the case. If no rule's condition is satisfied, the case is classified by means of the Nearest- Neighbor algorithm on exception cases (Exception_Case_Base).

3.2 Splitting Approaches

One of the most important problems in the present approach is to find a suitable Case_Base splitting procedure. In general this can be done by:

- *an expert* - the split obtained from an expert is very valuable (see result of experiments in section 4), and gives an opportunity to obtain an additional explanation for an exceptions.
- *a statistical approach* (e.g. cluster analysis) - this is a quite interesting approach from a formal point of view. Unfortunately, conventional statistic methods are mainly useful when the case is described by continuous features.
- *a heuristic approach* - there are a lot of informal approaches based on geometrical interpretations. The split can be based on weighting schemes as well, where exceptional cases can be determined to their performance and/or frequency of use for problem solving. This kind of method for identifying exceptional cases was introduced in Salzberg and Cost MVDM metric (Cost & Salzberg 1993).

We take into consideration two very simple heuristic split approaches. The first one assumes that non-typical cases can be interpreted as near-boundary cases. Aha's study on the IB2 algorithm (Aha 1992) shows that misclassified cases are more likely to be near-boundary, so in this heuristic correctly classified cases are put into the Standard_Case_Base, and incorrectly classified cases into the Exception_Case_Base. The second approach is more sophisticated, and is based on Zhang's formalization of the family resemblance idea (Zhang 1992). We assumed that standard cases have bigger intra-class similarity than inter-class similarity, and the opposite for exception cases. Intra-class similarity of a case is defined as a case's average similarity to other cases in the same class, and the inter-class similarity is defined as its average similarity to cases of all other classes. This kind of intra/inter similarity split heuristic we will call "Weak". The "Strong" one is when the standard cases have bigger intra-class similarity than the biggest inter-class similarity, that is computed for every other class separately.
Formally those heuristics can be defined as follows. There is a given set of cases: $C_{set}=\{c_1, c_2,...,c_N\}$. Each case c_j j=1..N is described by a list of attribute values $c_j=\langle f_j^1, f_j^2, \ldots f_j^m\rangle$. For each case $c_j \in C_{set}$ the classification: class(c_j) is given, that belong to the finite pre-numerated set. The task is to split C_{set} into two disjoint sets: C_{std} - set of standard cases (Standard_Case_Base), and C_{exc} - set of exception cases (Exception_Case_Base), $C_{set} = C_{std} \cup C_{exc}$. The similarity measure between two cases c_x and c_y is:

$$\mathrm{Sim}(c_x, c_y) = 1 - \sqrt{\frac{1}{m} \sum_{i=1}^{m} \mathrm{dis}(f_x^i, f_y^i)^2} \quad \text{where: } \mathrm{dis}(f_x^i, f_y^i) = \frac{\left|f_x^i - f_y^i\right|}{\max_i - \min_i}$$

for numeric- valued attributes, max_i, min_i respectively are the maximum and minimum value of the i-th attribute. For symbolic-valued attributes: if $f_x^i = f_y^i$ then $dis(f_x^i, f_y^i)=0$ (including both unknowns) else $dis(f_x^i, f_y^i))=1$. If one attribute value in unknown and the other is known, then they are of distance one. This similarity measure is reflexive, symmetrical,and normalized to the range from 0 to 1.

```
Begin
C_std ← C_exc ← ∅
For each c_x ∈ C_set
  Begin
   For each c_y ∈ {C_set - {c_x}}  sim(c_y) ← Sim(c_x,c_y)
   y_max ← max{sim(c_y)}
   If class(c_x)=class(y_max) Then C_std ← C_std ∪ {c_x} Else C_exc ← C_exc ∪{c_x}
  End
End
```

Fig.2. IB2 split heuristic

```
Begin
C_std ← C_exc ← ∅
For each c_x ∈ C_set
 Begin
  sim' ← sim'' ← 0
  For each c_y ∈ {C_set - {c_x}}
       If class(c_x)=class(c_y) then sim' ← sim' + Sim(c_x,c_y) else sim'' ← sim'' + Sim(c_x,c_y)
  intra_sim ← sim' / (| class(c_x) | -1)
  inter_sim ← sim'' / (N-| class(c_x) | )
  If intra_sim > inter_sim Then C_std ← C_std ∪ {c_x} Else C_exc ← C_exc ∪ {c_x}
 End
End

where: | class(c_j) | - number of cases from class: class(c_j)
```

Fig.3. Weak Inter/Intra similarity split heuristic

The first heuristic (based on IB2 algorithm) is presented in Fig.2, and the second one (Weak Intra/Inter similarity) is shown in Fig.3. The Strong Intra/Inter similarity heuristics can be easily obtained after slightly changing the code in Fig.3.

4 Experimental Results

The experiments were performed in order to compare the integrated approach based on the different split heuristic with rule based (rules from C4.5 algorithm), and the nearest- neighbor approach. The tests were based on three databases . The LED Display (with seven attributes) symbolic- valued artificial database with 10 % amount of noise. The U.S. Congressional Voting 1984, symbolic valued database with unknown values, and the Nurses (Surma 1994), a real database with numeric and symbolic attributes. Table 1 briefly characterises the domain and the experiments.

Table 1. Databases characteristics

Characteristic:	Database:		
	LED Display	**Voting** (1984)	**Nurses**
Train size	200	300	115
Test size	500	135	51
No of attributes	7	16	3
No of classes	10	2	5

All three databases were split into standard and exception subsets by means of IB2, Weak Intra/Inter similarity, and Strong Intra/Inter similarity heuristics. Additionally we obtain from the domain expert split for Nurses database. The characteristics of the knowledge base, respectively for Nearest- Neighbor (1-NN), C4.5 rules and Integrated approaches (Integration) are presented in Table 2. For example the knowledge base in the integration approach (with IB2 heuristics) for the database Voting consists of 26 exception cases and 4 rules. Those rules were generated from 274 standard cases (300 cases from Case_Base - 26 cases from Exception_Case_Base).

Table 2. Knowledge base sizes

Method:	Database:		
	LED display	**Voting** (1984)	**Nurses**
1-NN (no. of cases)	200	300	115
C4.5 rules (no. of rules)	15	7	9
Integration (IB2)	85 | 11	26 | 4	52 | 12
Integration (Strong)	54 | 12	34 | 6	61 | 4
Integration (Weak)	5 | 14	34 | 6	11 | 6
Integration (Expert)	-	-	39 | 6

Legend for Integration rows:
a | b ≡ no. of exceptions cases | no. of rules generated from standard cases

The results of the accuracy comparison are shown in Table 3. The results for Nearest- Neighbor (1-NN) were obtained thanks to the Inducer utility from the MLC++ Machine Learning Library in C++ (Kohavi et al. 1994). In all experiments the rules were generated and tested thanks to Quinlan's C4.5 Machine Learning programs (Quinlan 1993). For testing the integrated approach we created a special Case/Rule- Based System in the Kappa PC expert system shell.

Table 3. Average classification accuracies

Method:	Database:		
	LED display	**Voting** (1984)	**Nurses**
1-NN	69.3	94.1	56.9
C4.5 rules	69.9	95.6	60.8
Integration (IB2)	69.1	95.6	49.0
Integration (Strong)	69.7	97.0	58.8
Integration (Weak)	70.3	97.0	51.0
Integration (Expert)	-	-	66.7

To avoid overgeneralization, the rules for the integrated approach were induced based on standard and exception cases. But all exception cases had the same fictitious value of the decision variable. Next, the rules with fictitious variable were excluded from the obtained rule set. All the classifiers were tested on randomly drawn and separate training and test sets.
The accuracies for the integrated approaches on the LED with 10% noise are quite reasonable taking into account that LED consists of noisy or noisy free cases. The results for Voting are comparable with C4.5 rules, and much better than simple 1-NN. The Voting database has only 2 classes so the Weak Inter/Intra similarity split heuristic is equivalent to the "Strong" heuristic. If we compare split heuristics, it is easy to see that in all experiments results for the Inter/Intra similarity split are better than results for the IB2 split. The results on "Nurses" shows that the "Strong" split is much better than the "Weak" one. In this experiment all cases from the two classes were recognized by the "Strong" split as exceptions. The outstanding result was obtained for "Nurses", where the expert was responsible for splitting. These experimental results show that in terms of the accuracy the integrated approach is not worse than conventional approaches, and can give quite impressive outcomes for a suitable splitting procedure.

The explanatory ability of the integrated approach seems to be much better than the C4.5 rules or 1-NN. If the user requests an explanation, the system is showing a rule (if an input problem was interpreted as a standard) or a case (if an input problem was interpreted as an exception). The rules based on the standard cases are much more closer to the expert ones, than rules induced from the whole training set. In order to verify this hypothesis a simple experiment was done. According to the expert the Nurses set consists of 39 exceptions and 76 standard cases. From all 115 examples: 8 rules (R1) + a default rule were generated. Next, from 76 standard cases: 6 rules (R2) were generated as well. The rules from sets R1 and R2 were mixed and given to the expert for examination. The expert task was to evaluate each rule in terms of a nonsense, wrong, tolerable, or good rule. The result of this subjective evaluation is presented in Table 4.

Table 4. Expert evaluation of the "Nurses" rules

Rule set:	Expert evaluation:			
	"nonsense" (no. of rules)	**"wrong"** (no. of rules)	**"tolerable"** (no. of rules)	**"good"** (no. of rules)
R1 (based on 115 cases)	3	1	2	2
R2 (based on 76 cases)	0	2	2	2

The rules from the R2 set are much closer to the expert opinion than the rules from R1. In fact no rule from R2 was judged as a nonsense rule. Of course it is easy to falsify only one experiment, but this result opens up promising avenues for further evaluations.

5 Overview of the Learning Procedure

Unfortunately, even in such a simple integrated approach as is presented in this paper, the case retainment (learning) is very complex. In Fig.4 overview of the procedure of learning from failures is shown.

```
Begin
If  is(input_case, exception)
 Then
    Begin
      If solution_from_rules Then
                                            Begin
                                              specialize(Rule_Base, input_case)
                                              add(input_case, Exception_Case_Base)
                                            End
      If solution_from_exceptions And solution_is_false
                                            Then add(input_case, Exception_Case_Base)
    End
  Else { is(input_case, standard) }
    Begin
      If solution_from_exceptions Then  generalize(Rule_Base, input_case)
      If solution_from_rules And solution_is_false  Then modify(Rule_Base, input_case)
    End
End
where:
specialize(Rule_Base, case) - modify Rule_Base in order to not cover a case by any rule,
generalize(Rule_Base, case) - modify Rule_Base in order to cover a case by  at  least one rule
                                        and classify a case correctly,
modify(Rule_Base, case) - modify Rule_Base  in order to classify a case correctly.
```

Fig.4. Overview of learning from failures procedure.

We assumed that the case revision is done by asking an expert. In this process three sub- procedures on a Rule_Base are involved (i.e. generalize, specialize, and modify). In the conventional approach everyone of those sub- procedures needs an access to the whole Standard_Case_Base. This problem can be partially overcome by some incremental techniques (Utgoff 1989).

6 Concluding Remarks

The solution of the classification task that is proposed in this paper seems to be valuable for the three reasons. First, the knowledge acquisition process for obtaining Exception_Case_Base and Rule_Base is relatively easy. Second, the outcomes of the initial accuracy comparisons are acceptable, and very promising when splitting is based on the domain knowledge. Finally, a good comprehensibility of this approach is given to the end user.

The approach presented in this paper is one of the possible ways of integration. There are a lot of possibilities of integrating Case- Based and Rule- Based Reasoning. For instance a framework for integrating different integration strategies based on the NOOS object oriented language is presented in the mentioned Armengol and Plaza paper. Unfortunately the presented approach can be applied only where the underlying heuristic is appropriate for the domain.

Acknowledgements

We are very grateful to Eva Armengol, Eric Auriol, Maria Malek, and Enric Plaza for providing comments on the initial stage of this research. We also want to thanks P.M. Murphy and D.W. Aha who are in charge of the UCI Repository of Machine Learning Databases at the University of California (Irvine). Special thanks to Jerzy Stefanowski for outstanding support during the composition of this paper.

This research was undertaken with support from the European Commission's Phare ACE Programme under contract 94-0028-F

References

Aamodt, A. (1991). A Knowledge- Intensive, Integrated Approach to Problem Solving and Sustained Learning. *A Doctoral Dissertation* - University of Trondheim.

Aamodt, A. (1995). Knowledge Acquisition and Learning by Experience - The Role of Case-Specific Knowledge. In Kodratoff, Y., Tecuci, G. (eds.) *On Integration of Knowledge Acquisition and Machine Learning.* Academic Press (in press).

Aha, D.W. (1992). Tolerating noisy, irrelevant and novel attributes in instance- based learning algorithms. International Journal of Man Machine Studies, vol.36, pp.267-287.

Armengol, E., Plaza, E; (1994). Integrating Induction in a Case- Based Reasoner. In *Proceed. of the Second European Conference on Case- Based Reasoning.* AcknoSoft Press, Paris, pp.243-252.

Auriol, E., Manago, M., Althoff,K.D., Wess, S., Dittrich,S. (1994). Integration Induction and Case- Based Reasoning: Methodological Approach and First Evaluation. In *Proceed. of the Second European Conference on Case- Based Reasoning.* AcknoSoft Press, Paris, pp.145-156.

Bamberger, S.K., Goos, K. (1993). Integration of Case- Based Reasoning and Inductive Learning Methods. *In Proceed. of the First European Conference on Case- Based Reasoning*. SEKI Report SR-93-12. University of Keiserslautern, pp. 296-300.

Biberman, Y. (1995). The Role of Prototypicality in Exemplar-Based Learning. In Machine Learning: ECML-95, (Eds.) Lavrac, N., Wrobel, S., Springer Verlag, pp.77-91.

Barletta, R. (1994). A Hybrid Indexing and Retrieval Strategy for Advisory Case- Based Reasoning Systems Built with ReMind. In *Proceed. of the Second European Conference on Case- Based Reasoning*. AcknoSoft Press, Paris, pp.49-58.

Cardie, C. (1993). Using Decision Trees to Improve Case- Based Learning. In *Proceed. of the Tenth International Conference on Machine Learning*. Morgan Kaufmann, pp.25-32.

Cost, S., Salzberg, S. (1993). A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features. *Machine Learning* vol.10, pp.57-78.

Golding, A.R., Rosenbloom, P.S. (1991). Improving Rule- Based System through Case- Based Reasoning. In *Proceed. of the 1991 National Conference on AI*. The MIT Press, pp.22-27.

Hammond, K.J. (1988). Explaining and Repairing Plans That Fails. *Artificial Intelligence* vol.45, pp.173-228.

Kohavi, R., John, G., Long, R., Manley, D., Pfleger, K. (1994). MLC++. A Machine Learning Library in C++. In *Tools with Artificial Intelligence* Conference.

Koton, P.A. (1988). Reasoning about Evidence in Causal Explanations. In *Proceed. AAAI-88,* Morgan Kaufmann, Los Altos, pp.256-261.

Lenz, M. (1993). Cabata - A hybrid Case- Based Reasoning system. In *Proceed. of the First European Conference on Case- Based Reasoning*. SEKI Report SR-93-12. University of Keiserslautern, pp. 204-209.

Malek, M., Rialle, V. (1994). A Case- Based Reasoning System Applied to Neuropathy Diagnosis. In *Proceed. of the Second European Conference on Case- Based Reasoning.* AcknoSoft Press, Paris, pp.329-336.

Manago, M., Althoff, K.D., Auriol, E., Traphoner, R., Wess, S., Conruyt, N., Maurer, F. (1993). Induction and Reasoning from Cases. In *Proceed. of the first European Conference on Case- Based Reasoning*. SEKI Report SR-93-12. University of Keiserslautern, pp. 204-209.

Matwin, S., Plante, B. (1994). Theory Revision by Analyzing Explanations and Prototypes. In Michalski, R., Tecuci,G. (eds.) Machine Learning vol.4, Morgan Kaufmann, San Mateo.

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

Riesbeck, C.K., Schank, R.C. (1989). *Inside Case- Based Reasoning*. Lawrence Erlbaum, Hillsdale.

Rissland, E.L., Skalak D.B. (1991). CABARET: rule integration in a hybrid architecture. *International Journal of Man- Machine Studies*, vol.34, pp.839-887.

Surma, J. (1994). Enhancing Similarity Measure with Domain Specific Knowledge". In *Proceed. of the Second European Conference on Case- Based Reasoning.* AcknoSoft Press, Paris, pp.365-371.

Utgoff, P.E. (1989). Incremental Decision Tress. *Machine Learning,* vol.4, pp.161-186.

Zhang, J. (1992). Selecting Typical Instances in Instance- Based Learning. *In Proc. of the 9th Int. Conf. on Machine Learning.* Morgan Kaufmann, pp.470-479.