Multimodal Reasoning with Rule Induction and Case-Based Reasoning

Aijun An and Nick Cercone
Department of Computer Science, University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
e-mail : {aan,ncercone}@math.uwaterloo.ca

Abstract

The integration of problem-solving methods have attracted increasing research interest. We present a hybrid system, ELEM2-CBR, that integrates rule induction and case-based reasoning. ELEM2-CBR has the following characteristics. First, it applies a novel feature weighting function for assessing similarities between cases. By using this weighting function, optimal case retrieval is achieved in that the most relevant cases can be retrieved from the case base. Second, the method handles both classification and numeric prediction tasks under a mixed paradigm of rule-based and case-based reasoning. Before problem solving, rule induction is performed to induce a set of decision rules from a set of training data. The rules are then employed to determine some parameters in the new weighting function. For classification tasks, rules are applied to make decisions; if there is a conflict between matched rules, case-based reasoning is performed. In this paper, the new weighting function is presented. ELEM2-CBR's multimodal reasoning strategies are described. We demonstrate the performance of ELEM2-CBR by comparing its experimental results with the ones from other methods on a number of designed and real-world problems.

Introduction

Case-based reasoning (CBR) and rule induction (RI) are two major paradigms for machine learning. CBR involves reasoning from specific pieces (cases) of experience to solve new problems, while RI systems learn general knowledge from specific data and solve new problems by reasoning with the generalized knowledge. CBR and RI have complementary properties. CBR is good at learning continuous functions and nonlinearly separable categories, but it does not yield concise representation of concepts and is usually sensitive to irrelevant features. On the other hand, rule induction systems formulate comprehensible knowledge and can make effective use of statistical measures to detect relevant features. However, most RI systems only form axis-parallel frontiers in the instance space and have trouble in recognizing exceptions in small low-frequency sections of the space. In addition, due to the usual symbolic representation of rules, RI systems are not good at learning continuous functions.

The complementary properties of CBR and RI can be advantageously combined to solve some problems to which only one technique fails to provide a satisfactory solution. For example, CASEY [Koton 1989] adopts rule-based reasoning to perform case adaptation in which solutions to new problems are built from old solutions using the condition part of a rule to index differences and a transformational operator at the action part. Rules may also be used in similarity assessment by determining weights for features. INRECA [Althoff, Wess, and Traphoner 1995] builds a decision tree on the case base, weights of the features, with respect to the subclasses discovered in the tree, are computed, and class-specific similarity functions are defined based on these weights. On the other hand, several RI systems have employed CBR to make use of the information inherent in training examples to support their induction process. For example, CABARET [Skalak and Rissland 1990] uses CBR to aid a cooperating inductive decision-tree based learning algorithm with training set selection, branching feature selection, deliberate bias selection and specification of inductive policy. Another system, RISE [Domingos 1995], induces rules in a specific-to-general fashion, starting with a rule set that is a training set of examples. RISE examines each rule in turn, uses CBR to find the nearest example of the same class that it does not already cover and attempts to minimally generalize the rule to cover the class.

We present an integrated method, ELEM2-CBR, that combines CBR and RI to solve the following problems. First, an important issue in CBR is how to retrieve previous relevant cases from a case base. Determining a suitable set of weights for features is a significant task in case retrieval. There has been some research on design of feature weights, such as per-

category feature importance (PCF) and cross-category feature importance (CCF) methods [Creecy et al. 1992] (see [Wettschereck, Aha, and Mohri 1997] for a comprehensive review of feature-weighting methods). But optimality of most existing weighting methods remains. We propose an optimal weighting method, referred to as *relevance weighting*, that makes use of rule induction to assign weights to features. The relevance weighting method can achieve optimal case retrieval in that the most relevant cases can be retrieved. Secondly, ELEM2-CBR tackles the problem of learning continuous functions. Other integrated systems, such as INRECA [Althoff, Wess, and Traphoner 1995], RISE [Domingos 1995] and FCLS [Zhang 1990], focus on employing hybrid techniques to learn classification rules and therefore solve only classification problems. In ELEM2-CBR, since the RI technique is not good at learning continuous functions, CBR is used after rule induction to derive a numeric solution for the new problem from the retrieved relevant cases. Thirdly, ELEM2-CBR also solves classification problems. For classification problems, RI is used to learn major regularities of each concept, while CBR is used to classify examples in the boundary regions, i.e., small low-frequency sections of the example space.

In the rest of this paper, we first present the relevance weighting method. The multimodal reasoning strategies used by ELEM2-CBR for numeric prediction and classification are then described. An illustrative example for numeric prediction is provided. Finally we demonstrate the performance of ELEM2-CBR by comparing its experimental results with the ones from other methods on a number of designed and real-world problems.

Relevance Weighting

We define that a case in the case base is *relevant* to a new case if it is useful in solving the problem represented by the new case. Our objective in designing the relevance weighting method is to find weight settings for the features of a new case that allow the most relevant cases in the case base to be retrieved.

The Weighting Function

To achieve this objective, we propose using the following function, inspired by an idea of optimal document retrieval [Robertson and Sparck Jones 1976], to assign weights to each attribute-value pair of a new case as follows:

$$w(av_i) = \log \frac{p_i(1-q_i)}{q_i(1-p_i)}$$

where av_i is an attribute-value pair in the new case, p_i is the probability that av_i occurs in an old case in the case base given that the old case is relevant to the new case, while q_i is the probability that av_i occurs in an old case given that the old case is not relevant. It has been indicated [An 1997] that using this weighting function to assess the similarity between a new case and old cases in the training set yields a ranking of training cases in order of decreasing probability of relevance to the new case. Thus, optimal case retrieval is achieved in terms that the most probably relevant cases can be retrieved. Suppose that there are N cases in the case base of which R cases are relevant, and the attribute-value pair av_i occurs in n cases, of which r cases are relevant. Using simple proportion estimations of the two probabilities, the formula becomes:

$$w(av_i) = \log \frac{r(N-n-R+r)}{(n-r)(R-r)}$$

Estimation of Parameters Using RI

To use the above weighting function, parameters N, R, n and r need to be determined. Obviously, N and n are easy to be obtained. However, R and r, i.e, the information about which cases or how many cases in the case base are relevant, are normally not available in advance. Therefore, a method for estimating these two parameters is needed.

We propose to use a RI technique to estimate R and r. Before CBR is performed, RI is applied to derive rules from the case base. When a new case is presented, it is matched with the rules. If there is a single match, i.e., only one rule is matched with the new case, or if there are multiple matches but the matched rules predict the same concept, then the cases that belong to the concept indicated by the matched rule(s) are considered relevant to the new case. If there are multiple matches and the matched rules indicate different concepts, then the new case is in the boundary region between the indicated concepts. In this situation, all the cases that belong to the indicated concepts are considered relevant. In the case of no-match, i.e, no rules are matched with the new case, partial matching is performed in which some attribute-value pairs of a rule may match the values of corresponding attributes in the new case. A partial matching score between the new case and a partially matched rule is calculated. The concepts that are indicated by partially matched rules compete each other based on their partial matching scores. The cases that belong to the concept that wins the competition are chosen as relevant cases. After the set S of relevant cases is determined, R is assigned as the number of the cases in S and r is set to the number of cases in S that match the attribute-value pair av_i.

Numeric Prediction

If the task is to predict numeric values, problem solving in ELEM2-CBR is basically a CBR process which involves case retrieval and case adaptation. RI is used during retrieval for relevance weighting. After weights are determined, ELEM2-CBR calculates the similarity between each case x in the case base and a new case q:

$$Similarity(x, q) = \sum_{i=1}^{n} w_i \times Simil(x_i, q_i)$$

where n is the number of attributes, x_i is x's value for the ith attribute a_i, q_i is q's value for a_i, w_i is the weight for q's ith attribute-value pair calculated using the new relevance weighting method, and

$$Simil(x_i, q_i) = \begin{cases} 0 & \text{if } a_i \text{ is symbolic and } x_i \neq q_i; \\ 1 & \text{if } a_i \text{ is symbolic and } x_i = q_i; \\ 1 - |n(x_i) - n(q_i)| & \text{if } a_i \text{ is continuous.} \end{cases}$$

where $n(x_i)$ and $n(q_i)$ denote the normalized values of x_i and q_i respectively. We can say that if cases are ranked in order of decreasing value of their similarity to the new case, the ranking is actually a ranking of cases in order of their decreasing probability of relevance to the new case [An 1997]. Thus, the most relevant cases can be retrieved.

After case retrieval, case adaptation is performed as follows:

1. Select a set S of k most relevant cases to the new case q where k is a user-defined parameter.
2. For each case c_i in S, compute a partial contribution value of c_i as

   $$PCV(c_i, q) = Similarity(c_i, q) \times F(c_i)$$

   where $F(c_i)$ is the real decision value of case c_i that is stored in the case base.
3. Let $Sum = \sum_{c_i \in S} Similarity(c_i, q)$.
4. Compute a numeric decision value for the new case q as:

   $$Prediction(q) = \frac{\sum_{c_i \in S} PCV(c_i, q)}{Sum}.$$

Classification

If the application task is to classify a new case into a category, ELEM2-CBR performs the classification by using both rule-based and case-based reasoning. Rule-based reasoning is conducted first. If there are matches between rules and the new case and if the matched rules provide a unanimous decision, the new case is classified by the matched rule(s). Otherwise, case-based reasoning is conducted to solve the conflicts between rules or to deal with other situations as follows. If conflicts exist between matched rules, or if there is no match, i.e, no rules are matched with the new case, but partial matches exist, then rank the cases in the case base by using the relevance weighting method and the similarity measure described in the last section. If partial matches do not exist, then rank the cases in the case base using the weighting function with $R = r = 0$ and the similarity measure described in the last section. After cases are ranked, select a set S of k most relevant cases from the ranked cases where k is a user-defined parameter. If all the cases in S predict class C, the new case is classified into C. Otherwise, for each class Y_i that exists in S, compute a decision score of Y_i defined as:

$$DS(Y_i) = \sum_{j=1}^{m} Similarity(c_j, q)$$

where c_j is one of the m cases in S that predict Y_i and q is the new case. The new case is classified into the concept that has the highest decision score.

An Example for Numeric Prediction

We have applied ELEM2-CBR to daily water demand prediction. In this application, a set of previous cases during a period of 3 years is collected. The cases are represented in terms of a number of attributes, such as temperatures, humidity, day of the week, etc. Figure 1 shows an example of this application in which, given a case base and a new case, a set of rules are induced, a weight setting is calculated, relevant cases are retrieved and finally a solution for the new case is derived. Results for this application have been presented in [An, Chan, and Cercone 1997].

Empirical Evaluation

ELEM2-CBR has been evaluated by comparing its performance with other methods on designed and real-world problems. We implemented ELEM2-CBR and three other case-based reasoning algorithms, referred to as CBR-NW, CBR-PC and CBR-CC. These three algorithms are similar to ELEM2-CBR except that no rule induction is performed in the three algorithms and different weighting methods are used. CBR-NW considers the attributes are equally important, so it assigns equal weights to every attribute. CBR-PC employs the PCF weighting method and CBR-CC uses the CCF method (see [Creecy et al. 1992] for descriptions of PCF and CCF). In our experiments, the four

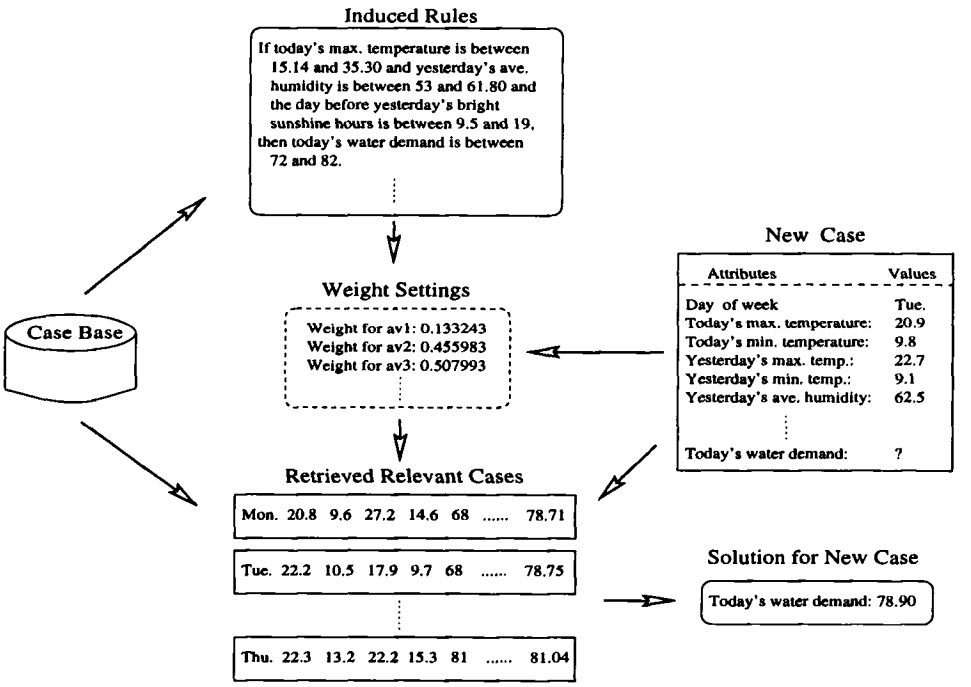

Figure 1: Illustration of ELEM2-CBR for Predicting Water Demand

algorithms run in an incremental learning mode.[1] On some classification problems, we also compare ELEM2-CBR with a pure rule induction system, ELEM2, which ELEM2-CBR has evolved from. ELEM2 [An 1997] generates classification rules by selecting the most relevant attribute-value pairs. It employs several techniques, such as post-pruning of generated rules and probabilistic classifications, to deal with possible noise in the training data.

Performance of a tested algorithm on classification problems is measured by classification accuracy, i.e., the percentage of correct classifications made by the algorithm on a set of test examples. On numeric prediction problems, performance is measured by the average of relative errors over test examples.

Experiments on Classification Problems

Five problems are designed to evaluate the performance of ELEM2-CBR on classification problems. Each problem contains a target concept, i.e., an example in a problem either belongs to the target concept or does not belong to it.

Problem *P1* contains five nominal conditional attributes with four values each: 0, 1, 2, and 3. The target concept is "if and only if any two or more of the first three attributes of an example have value 0 or 1, then the example belongs to the concept". From the entire space of 1024 possible examples, 256 were randomly chosen as training examples and the remaining as the testing set.

Problems *P2* and *P3* are designed to test ELEM2-CBR's ability to learn concepts with non-linear boundaries. Each problem contains two continuous attributes representing two axes (x and y) in a two dimensional space. An irrelevant attribute is added to each problem to test the algorithms' ability to tolerate irrelevant features. The target concepts of *P2* and *P3* are "if $ax^2 + by^2 \leq c$, then the example belongs to the concept" and "if $y > ax^3 + bx^2 + cx + d$, then the example belongs to the concept", respectively, where a, b, c and d are constants.

Problem *P4* is the same as *P3* except that there is no irrelevant feature in the data. Problem *P5* is derived from *P4* by randomly adding 5% classification noise into the training set. For each problem, a set of examples is chosen from the instance space, one-third of which are used as the training set and the reminder constitutes the testing set. The results of the experiments on each problem in terms of classification accuracy on testing sets are shown in Table 1. The best result for each problem is highlighted in boldface.

From the results, we can see that ELEM2-CBR, ELEM2 and C4.5 perform perfectly on problem *P1*, while the three pure CBR algorithms do not perform

[1] By incremental learning we mean the already tested examples in the testing set are used in case-based reasoning (not in rule induction) to solve problems represented by later test cases.

Problem	CBR-NW	CBR-PC	CBR-CC	ELEM2-CBR	ELEM2
P1	85.55%	64.84%	84.64%	**100%**	**100%**
P2	93.40%	70.40%	96.00%	**98.60%**	96.60%
P3	92.86%	74.06%	86.09%	**95.86%**	95.13%
P4	**96.24%**	83.46%	87.22%	95.86%	95.50%
P5	95.12%	76.32%	76.70%	**95.16%**	94.00%
AVERAGE	92.63%	73.82%	86.13%	**97.10%**	96.25%

Table 1: Performance Comparison on Designed Classification Problems.

Dataset	CBR-NW	CBR-PC	CBR-CC	ELEM2-CBR
australian	85.07%	**86.67%**	80.43%	85.65%
breast-cancer	**96.93%**	94.58%	96.78%	95.90%
glass	71.96%	52.34%	51.87%	**74.77%**
heart	81.48%	79.63%	80.37%	**82.22%**
tic-tac-toe	67.43%	65.34%	75.26%	**99.37%**
zoo	94.06%	58.42%	94.06%	**96.04%**
AVERAGE	82.82%	72.83%	79.80%	**88.99%**

Table 2: Performance Comparison on Real-World Classification Problems

well. This is because the concept in *P1* has "rectangular" boundary regions and rule induction algorithms are good at learning and representing this kind of concepts, while pure CBR algorithms are not . On the other four problems, ELEM2-CBR performs better than pure rule induction algorithm: ELEM2. This result is consistent with what we expected: rules are not good at representing concepts with non-linear boundaries. On *Problem 4*, CBR-NW performs the best among the algorithms. The reason for this is that there is no irrelevant feature or noise in this problem and the two features are equally important. In addition to artificial domains, we have also experimented with 6 real-world datasets from the UCI repository, for which the underlying concepts are unknown. Table 2 reports the results of *leave-one-out* evaluation on the 6 datasets.

Experiments on Numeric Prediction Problems

To evaluate ELEM2-CBR's ability to predict numeric values, we have conducted experiments with CBR-CC, CBR-PC, CBR-CC and ELEM2-CBR on four designed numeric prediction problems and three real-world problems from the UCI repository. Definitions of the designed problems are as follows:

$$\begin{aligned} NP\text{-}1: \quad f(x,y,z) &= x^2 + y^2 + z^2 \\ NP\text{-}2: \quad f(x,y) &= \log_e(x) + \log_e(y) \\ NP\text{-}3: \quad f(x,y) &= sin^{-1}(x) + cos^{-1}(y) \end{aligned}$$

NP-4 is derived from *NP-1* by randomly adding 5% prediction noise into the training set. For each problem, a set of examples are picked up from the domain.

Problem	CBR-NW	CBR-PC	CBR-CC	ELEM2-CBR
NP-1	2.36%	8.18%	2.49%	**2.02%**
NP-2	1.74%	2.68%	**1.45%**	1.69%
NP-3	5.92%	26.00%	11.57%	**5.77%**
NP-4	2.50%	8.01%	2.65%	**1.84%**
AVERAGE	3.13%	11.22%	4.54%	**2.83%**

Table 3: Performance Comparison on Designed Numeric Problems.

Problem	CBR-NW	CBR-PC	CBR-CC	ELEM2-CBR
housing	**12.55%**	19.82%	17.29%	12.81%
imports-85	12.06%	21.24%	15.18%	**11.72%**
machine	44.77%	66.11%	55.10%	**37.00%**
AVERAGE	23.13%	35.72%	29.19%	**20.51%**

Table 4: Performance Comparison on Actual Numeric Problems.

One-third of them are randomly selected as training examples and the remaining ones as test samples. For each problem, the average of the relative errors made by each tested algorithm over the testing examples is reported in Table 3. Boldface is used to indicate the best result on each problem. The results of *leave-one-out* evaluation on three selected real-world datasets, *housing, imports-85*, and *machine*, are shown in Table 4.

Conclusions

We have presented the multimodal reasoning strategies used in ELEM2-CBR. In ELEM2-CBR, RI assists CBR in determining parameters for an optimal weighting function, while CBR assists RI in learning continuous functions and classifying boundary examples. ELEM2-CBR has been evaluated empirically on both designed and actual databases. The results show that ELEM2-CBR gives better predictive precision than other three CBR programs with different weighting methods. ELEM2-CBR is also compared favorably to a pure rule induction system on designed classification problems.

Acknowledgements

This research is supported by Natural Science and Engineering Research Council of Canada (NSERC). We would also like to thank Telecommunications Research Laboratories (TRLabs) for its previous support to this research.

References

Althoff, K., Wess, S. and Traphoner, R. 1995. INRECA - A Seamless Integration of Induction and Case-Based Reasoning for Decision Support Tasks. In Pro-

ceedings of the 8th Workshop of the German Special Interest Group on Machine Learning.

An, A. 1997. Analysis Methodologies for Integrated and Enhanced Problem Solving. Ph.D. Dissertation, Dept. of Computer Science, University of Regina, Canada.

An, A., Chan, C., and Cercone, N. 1997. Water Demand Forecasting Using Case-Based Reasoning. In Proceedings of IJCAI'97 Workshop on Practical Use of Case-Based Reasoning, Nagoya, Japan. 99-110.

Creecy, R.H., Masand, B.M., Smith, S.J. and Waltz, D.L. 1992. Trading MIPS and Memory for Knowledge Engineering. *Communications of the ACM*, 35: 48–64.

Domingos, P. 1995. Rule Induction and Instance-Based Learning: A Unified Approach. In Proceedings of IJCAI-95. 1226–1232. Montreal, Canada.

Koton, P. 1989. Using Experience in Learning and Problem Solving. Ph.D. Dissertation, Laboratory of Computer Science, Massachusetts Institute of Technology. MIT/LCS/TR-441.

Robertson, S.E. and Sparck Jones, K. 1976. "Relevance Weighting of Search Terms". *Journal of the American Society for Information Science*, 27: 129–146.

Wettschereck, D., Aha, D.W. and Mohri, T. 1997. A Review and Empirical Evaluation of Feature Weighting Methods for a Class of Lazy Learning Algorithms. *Artificial Intelligence Review*, 11: 273–314.

Zhang, J. 1990. A Method That Combines Inductive Learning with Exemplar-Based Learning, In Proceedings of the 2nd International IEEE Conference on Tools for Artificial Intelligence, Herndon, VA. 31–37.