Rule-Induction and Case-Based Reasoning: Hybrid Architectures Appear Advantageous

Nick Cercone, *Senior Member, IEEE*, Aijun An, and Christine Chan

Abstract—Researchers have embraced a variety of machine learning (ML) techniques in their efforts to improve the quality of learning programs. The recent evolution of hybrid architectures for machine learning systems has resulted in several approaches that combine rule-induction methods with case-based reasoning techniques to engender performance improvements over more-traditional one-representation architectures. We briefly survey several major rule-induction and case-based reasoning ML systems. We then examine some interesting hybrid combinations of these systems, and explain their strengths and weaknesses as learning systems. We present a balanced approach to constructing a hybrid architecture, along with arguments in favor of this balance and mechanisms for achieving a proper balance. Finally, we present some initial empirical results from testing our ideas and draw some conclusions based on those results.

Index Terms—Case-based reasoning, rule induction, machine learning, classification, numeric prediction.

1 Introduction

Machine learning (ML) has evolved rapidly over the past two decades. ML researchers have embraced a variety of machine learning techniques in their efforts to improve the quality of learning programs. The relatively recent development of hybrid representations for ML systems have resulted in several interesting approaches which combine rule-induction (RI) methods with case-based reasoning (CBR) techniques to engender performance improvements over more traditional one representation architectures.

CBR is used in learning and problem-solving systems to solve new problems by recalling and reusing specific knowledge obtained from past experience. RI systems learn general domain-specific knowledge from a set of training data and represent the knowledge in comprehensible form as IF-THEN rules. RI systems also often succeed in identifying small sets of highly predictive features, and can make effective use of statistical measures to eliminate noise in data.

We present a brief survey of several major rule-induction and case-based reasoning ML systems that have been deployed. We point out some idiosyncrasies of these systems. We then examine some interesting hybrid combinations of these systems and explain their strengths and weaknesses as learning systems. Due to their complementary properties, CBR and RI techniques have been combined in some systems to solve problems to which a single technique fails to provide a satisfactory solution. We explain how RI can be used in a supportative role in ML systems that employ CBR and, likewise, how CBR has been used in a supportative role in systems employing RI as the major learning paradigm. We present a balanced approach to constructing an integrated CBR and RI hybrid architecture, along with arguments in favor of this balance and mechanisms for achieving a proper balance. The key to making the balance work properly is a new weighting function which helps select relevant cases from a case base expeditiously. *Relevance weighting* assesses similarities between cases, making use of RI results to assign weights to each attribute-value pair of the query case. Cases can then be ranked according to their probability of relevance to the new case, thus producing the most possibly relevant cases for retrieval. Our method, known as ELEM2-CBR, performs both classification and numeric prediction under a mixed RI and CBR paradigm.

Finally, we present some initial empirical results derived from testing our ideas, and we draw some conclusions based on those results.

2 Survey

Two primary goals of machine learning are to understand and model human learning behavior and, more pragmatically, to provide increasing levels of automation in the knowledge acquisition process. United by common goals, ML research has emphasized different approaches. RI, neural networks, genetic algorithms, analytic learning, Bayesian learning, reinforcement learning, and CBR have emerged as ML paradigms. Research into neurobiology (neural nets), evolution theories (genetic algorithms), formal logic (analytic methods), heuristic search (rule-induction) and human memory (case-based reasoning) have provided researchers with analogical models to study. Langley's review [16] describes applications of rule induction, the most widely studied methodology.

- *N. Cercone and A. An are with the Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1. E-mail: {ncercone, aan}@uwaterloo.ca.*
- *C. Chan is with the Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada S4S 0A2. E-mail: chan@cs.uregina.ca.*

Manuscript received 16 May 1997; revised 13 Aug. 1998.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 108304.

Representing knowledge in comprehensible condition-action rules, general domain-specific knowledge is learned from a set of training data in RI. Most RI systems conduct heuristic search through the hypothesis space of rules or decision trees. RI systems typically use a statistical evaluation function to select attributes or attribute-value pairs for incorporation into the knowledge structure. Pre- or postpruning of the knowledge structure is usually conducted to handle imperfect or noisy training data. Many RI systems have been applied to real-world domains to discover knowledge from observed data. Example systems include C4.5 [21], AQ15 [18], and CN2 [10]. Clark [9] provides an overview of RI techniques and strategies for noise abatement. Despite their successes, RI systems have stood accused of forming only hyper-rectangular regions in the example space and not recognizing exceptions in small, low frequency sections of the space. Furthermore, rules do not represent continuous functions well.

CBR represents knowledge by storing descriptions of previously experienced, specific cases. A new case is solved by retrieving similar past cases and adapting their solutions. Common retrieval schemes employ variations of the nearest neighbor method in which similarity metrics are used to identify cases nearest to the current case. An overview of the CBR foundational issues is presented in Aamodt [1]. Although CBR is a relatively recent learning and problem-solving method, a number of commercial tools have been developed. Watson [26] provides a review of available CBR tools.

CBR can learn nonlinearly separable categories of continuous functions and CBR is incremental by nature, unlike most inductive learning methods which have difficulty extending or refining their rule set during the problem-solving stage. CBR, however, does have limitations: It does not yield concise representations of concepts which can be understood easily by humans and CBR systems are usually sensitive to noise.

2.1 Hybrid Approaches

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. Generally the combination involves CBR systems using rule-based reasoning for support. CBR can also be used in a support role or integrated with rule-based reasoning in some balanced fashion.

CBR processing can be augmented with rule-based techniques when general domain knowledge is required. For example, adaptation tasks in the CBR processing cycle are usually performed by rule-based systems where the rules capture a theory of case adaptation and the necessary aspect of the domain theory to carry out the changes [17]. CASEY [15] is a system where case adaptation is performed by rule-based reasoning in which solutions to new problems are built from old solutions using rules with the condition-part indexing differences and with a transformational operator as the action part. Rules can also be used to guide the search and matching processes in retrieval tasks of a CBR system. Rules regarding the problem domain may serve to organize the case base and, when applied, focus the search space to more relevant cases. Rules may also be used in similarity assessment by determining weights for attributes. INRECA [2] builds a decision tree on the case database, weights of the attributes, with respect to the subclasses discovered in the tree, are computed, and class specific similarity functions are defined based on these weights. Rule-based reasoning can aid case retrieval by justifying a candidate set of cases as plausible matches, e.g., knowledge-based pattern matching (rule-based reasoning) is used in PROTOS [6] to confirm new case expectations.

CBR can also serve in a supporting role. Unlike rules, cases in a case base contain specific knowledge about a domain. When general domain knowledge is not accessible, the specific knowledge inherent in cases can provide valuable information to solve problems. Because CBR can elicit domain knowledge through its analysis of cases, CBR can aid systems with tasks where general domain knowledge is not available but needed.

Several RI systems have employed CBR to make use of the information inherent in training cases support their induction process. CABARET [25] 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. CBR is used to form categories of a training set which include most-on-point cases, best cases, near miss cases, trumping cases, and conflict cases. These case taxonomies allow the learning system to consider the various roles cases play in addition to classification, say, as positive or negative examples. For feature selection, CABARET takes advantage of CBR-provided domain knowledge as well as information-theoretic methods to select branching attributes for growing decision trees. RISE [10] induces rules in a specific-to-general fashion, starting with a rule set that is the 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.

Balanced combination techniques use CBR and rule-based techniques to support each other in a learning, problem-solving environment, neither of which is in a purely support role. Example systems include INRECA [2], FCLS [27], and ANAPRON [14]. INRECA performs classification by first generating and trying a decision tree, generated from the case base, to navigate the search for a matched or similar concept. The generalized knowledge is also used to improve retrieval by determining attribute weights (degree of attribute importance for similarity case measurement) with respect to the subclasses discovered by the decision tree. If INRECA can answer a given query at this point, no further action is required, otherwise their hybrid approach applies CBR when the query lies outside the induced concept region.

FCLS and ANAPRON focus on hybrid representations of a concept. In both approaches, a concept is represented by two parts: A generalized abstract description in the form of rules and a set of exceptions in the form of exemplars.[1] Since rules represent broad domain trends and cases usefully “fill in” rule exceptions, a hybrid approach is supported. Rules and exceptions are generated with an inductive learning algorithm in FCLS. Rules are induced according to some criteria; when no acceptable rules can be generated for a concept, a set of exemplars are selected as exceptional cases of the concept. Both rules and exemplars are used to match the new case during problem-solving. Problem-solving in Golding et al.’s system works differently. Their system applies rules to the target problem to approximate the answer. However, if the problem is judged to be compellingly similar to a known exception to the rules in any aspect of its behavior, then the aspect is modified after the exception rather than the rule.

INRECA’s advantage lies in its incremental learning of decision trees. Over time, more and more generalized concepts can be induced based on the increasing case base. Thus INRECA evolves from a more or less pure CBR system to a system based on inductively learned knowledge. INRECA does not address uncertainty, i.e., when a new case is in the boundary region of two or more concepts and thus is covered by rules that belong to different concepts. FCLS addresses such conflict resolution in its hybrid representation by computing a degree of fit between a new case and a rule or an exemplar. FCLS, moreover, sets the feature weights for exemplars equally, which is not desirable in most situations. Furthermore, FCLS has only a weak ability to deal with noise. For noisy training examples, small disjuncts (used by FCLS as exemplars) may be indicative of data errors. Differentiating noise and boundary examples is not handled in FCLS; Golding’s ANAPRON has similar problems.

3 Integrating Rule-Induction (RI) and Case-Based Reasoning (CBR)

We propose a new hybrid method which integrates RI and CBR techniques. Our ELEM2-CBR employs relevance weighting to access similarities between cases, making use of RI results to assign weights to each attribute-value pair of the query case. Cases in the case-base can then be ranked according to their probability of relevance to the new case. ELEM2-CBR performs classification and numeric prediction under a mixed paradigm of rule-based and case-based reasoning. After performing RI, induced rules are applied in case retrieval to determine weight settings for features and to detect noise in the training set for removal before CBR is conducted. During classification, rules are applied to make decisions; conflicts observed between matched rules are resolved by performing CBR.

3.1 Weighting and Parameter Estimation

Feature weighting is a key issue in case retrieval. Many case-based reasoning algorithms retrieve cases using the k-nearest neighbor (k-NN) method with a weighting methods, such as the per-category feature (PCF) and cross-category feature (CCF) importance measures [11]. Although PCF and CCF can be clearly calculated, optimality determination (in any sense) remains. Furthermore, PCF and CCF are case specific and do not take into account the query case when assigning weights to features. Our proposed method, relevance weighting, overcome these objections.

Cooper [8] proposed a probability ranking principle (PRP) for information retrieval which we adapt:

> “If a retrieval system’s response to each request is a ranking of the documents in the collection in order of decreasing probability of usefulness to the user, then the overall effectiveness of the system to its users will be the best that is obtainable on the basis of that data.”

Robertson [23] provided formal justification proving that PRP leads to optimal performance in terms of retrieval effectiveness and that PRP is the correct decision procedure to use. Robertson and his colleagues also built an information retrieval system, named OKAPI [24], based on PRP. The performance of this system has been ranked at the top places in the Text REtrieval Conference (TREC), during which a large scale experiment/competition involving a number of research groups working on test retrieval is conducted [7], [24].

Considering that PRP has been both theoretically and experimentally proven to be a correct decision procedure for retrieval and that CBR and information retrieval share similar goals, it is natural to invoke the PRP principal for case retrieval. We speculate 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. For optimal case retrieval, we restate the PRP as:

> “*The probability ranking principle for optimal case retrieval*: If a case retrieval system’s response to a new case (query) is a ranking of cases in the case base in order of decreasing probability of query relevance, where the probabilities are estimated as accurately as possible on the basis of whatever data has been made available to the system for the purpose of using the retrieval result to solve the problem represented in the query, then the overall effectiveness of the system in terms of the probability of relevant cases being retrieved will be the best that is obtainable on the basis of that data.”

By this principle we claim that optimal case retrieval can be achieved if the system ranks the retrieved cases in the decreasing order of their probability of relevance to the new case, thus maximizing the probability of relevant cases being retrieved. To set weights for optimal retrieval, let p denote the probability that a query term t occurs in a document, given that the document is relevant; let q be the corresponding probability for a nonrelevant document. Robertson and Spark Jones [22] show that assigning weights to query terms with values of

$$\log \frac{p(1-q)}{q(1-p)}$$

[1] Exceptions have different meanings in the two systems. In FCLS, an exception represents a rare concept case, while in Golding’s system an exception represents an example that does not agree with a rule (exception of a rule).

yields an optimal result[2] if the terms are mutually independent.[3] Using this formula to assign weights to query terms results in a ranking of documents that leads to optimal document retrieval.

To achieve optimal case retrieval in CBR, suppose that a query case *qc* in CBR consists of a set of attribute-value pairs $\{av_1, av_2, \cdots, av_n\}$, where n is the number of attributes. The role of attribute-value pairs av_i ($i = 1, \cdots, n$) in case retrieval is the same as the role of query terms in document retrieval, thus we assign weights to attribute-value pairs of *qc* as follows

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

where 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. We have assumed symbolic attributes; discretization is performed to transform a continuous attribute domain into symbolic ranges.[4]

When cases in the case base are known as relevant or not relevant, we calculate the weight as

$$w(av_i) = \log \frac{r(N-n-R+r)}{(n-r)(R-r)} \tag{1}$$

where 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. Although N and n are easy to obtain, R and r are not normally available in advance. Thus a method is needed to estimate these two parameters to determine relevance weighting.

For classification problems, we can assume that every case in the case base belongs to a symbolic concept. Furthermore, cases relevant to a new case, i.e., useful for solving the problem represented by the new case, are those that belong to the same concept as the new case. Thus, if we can estimate to which concept the new case belongs, then the cases in the case base that belong to the concept are considered relevant to the new case. RI systems can analyze data and generate classification rules from the data. In addition, RI systems can make effective use of statistical measures to detect noise and irrelevant features. We use RI and deduction to estimate the parameters R and r. ELEM2 [3], [4], [5] performs RI and uses several techniques, including post-pruning of generated rules and probabilistic classifications to eliminate noise in the training data.

ELEM2 is applied to derive rules from the training cases. When a new case is presented, it is matched with the rules. If there is only one rule 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. Multiple matches where the matched rules indicate different concepts indicate that the new case is on the boundary region between the indicated concepts; in this situation all cases belonging to the indicated concepts are considered relevant. When no rules are matched with the new case, partial matching is performed to determine whether some attribute-value pairs of a rule match the corresponding attributes in the new case. A partial matching score is calculated between the new case and a partially matched rule. Concepts indicated by partially matched rules compete with each other based on these scores and 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 cases in S and r is set to the number of cases in S that match the attribute-value pair av_i.

We now wish to use the weighting function to assign scores to training cases to rank them in order of probability of relevance. If all features are symbolic, given a new case q, weights for each attribute-value pair of q are calculated according to (1). For each case x in the case base, we assign a score of x as the sum of the weights of those attribute-value pairs in q that occur in x. If cases are ranked in decreasing value of this score, then the ranking is actually a ranking of the cases in order of their decreasing probability of relevance to the new case. For cases containing continuous attributes, we adjust the weight for a continuous attribute by multiplying the absolute difference between q's value for that attribute and x's value for the same attribute. Thus the function to calculate the score for a case x becomes a similarity measurement between x and q, and is stated as

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

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

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

[2] In their proof, the retrieval effectiveness is measured by the following probabilities:

$$recall = P(\text{document retrieved} \mid \text{document relevant});$$
$$fallout = P(\text{document retrieved} \mid \text{document nonrelevant});$$
$$precision = P(\text{document relevant} \mid \text{document retrieved}).$$

Another parameter is:

$$\phi(d_i) = P(\text{document relevant} \mid \text{document is } d_i) = P(d_i \text{ is relevant}).$$

Robertson and Spark Jones proved that, given expected *fallout*, *recall* can be maximized by including in the retrieved set those documents with the highest values of $\phi(d_i)$. In other words, optimized retrieval can be obtained by ranking documents in order of $\phi(d_i)$ and applying a cut-off when the given *fallout* is reached. They also claimed that expected *fallout* is minimized for given expected *recall*, or that expected *precision* is maximized for given expected *recall* or given expected *fallout*, using the same document ordering.

[3] In practice, query terms may not be mutually independent. However, research has shown that consideration of term dependencies results in a much more complicated weighting function which requires many parameters to be estimated. But the retrieval performance in terms of precision and recall improves very little by using the complicated function [24]. Therefore, we only adopt the simple version of the weighting function, which has the independence assumption.

[4] We use a discretization method that is similar to the one reported in [13], in which an entropy minimization heuristic is used to discretize the range of a continuous-valued attribute into multiple intervals.

where $norm(x_i)$ and $norm(q_i)$ denote the normalized values of x_i and q_i, respectively, and $|norm(x_i) - norm(q_i)|$ denotes the absolute value of $norm(x_i) - norm(q_i)$.

3.2 Problem Solving in ELEM2-CBR

ELEM2-CBR employs the weighting and case ranking methods discussed above and can perform both classification and numeric prediction. Given a set of training data, ELEM2-CBR performs RI using ELEM2 to generate a set of classification rules for both tasks. ELEM2's classification is performed over a set of training data after RI and misclassified cases, considered as noise, are removed from the case base, before CBR is performed.

If the task is to predict numeric values, problem solving in ELEM2-CBR is basically a CBR process. Relevance weighting and the case ranking methods discussed are employed in case retrieval. Rules generated by ELEM2 are used to determine parameters in the weighting function. After ranking cases in the case base according to relevance to the new case, several of the most relevant cases are selected and their solutions adapted to the new case. Our adaptation procedure is simple:

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:

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

ELEM2-CBR performs both deductive reasoning and CBR for classification tasks, i.e., when the task is to classify a new case into a category, as follows:

1) Match the new case with the rules generated by ELEM2.
2) If there is a single match, i.e., only one rule is matched with the new case, then the case is classified into the class that the rule indicates.
3) If there are multiple matches, but the matched rules indicate the same class C, then the new case is classified into C.
4) If there are multiple matches and the matched rules indicate different classes, 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 weighting method and the similarity measure described in the last section. The parameters in the weighting function are determined by considering as relevant cases those cases that belong to the classes that matched rules (or partially matched rules in the case of no match) indicate. Go to step 6.
5) 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.
6) Select a set S of k most relevant cases from the ranked cases where k is a user-defined parameter.
7) If all the cases in S predict class C, the new case is classified into C.
8) 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.
9) Classify the new case into the concept that has the highest decision score.

Steps 1-3 in this procedure perform deductive reasoning to classify the new case. CBR is conducted to resolve conflicts between rules or to deal with partial matching. Steps 4-6 perform case retrieval to determine the set of relevant cases needed by the relevance weighting function. Steps 7-9 perform case adaptation to determine the new case's class from the retrieved cases. Fig. 1 illustrates the integration of CBR and rule-based techniques in ELEM2-CBR.

4 Empirical Evaluation

We investigate whether ELEM2-CBR's expected benefits are observed in practice. We compare ELEM2-CBR with three other case-based reasoning algorithms: CBR-NW, CBR-PC, and CBR-CC, which are similar to ELEM2-CBR but without an RI part. CBR-NW assigns equal weight to every attribute; CBR-PC and CBR-CC employ the PCF and CCF weighting methods, respectively. The programs of CBR-NW, CBR-PC and CBR-CC are run in an incremental learning mode,[5] while ELEM2-CBR is not. We also compare ELEM2-CBR with C4.5 [21] and OC1 [20] on classification problems. Both C4.5 and OC1 are decision tree learning systems. C4.5 learns axis parallel decision trees in which tests at each node are equivalent to axis-parallel hyperplanes in the attribute space, while OC1 builds both oblique and axis parallel decision trees that test a linear combination of the attributes at each internal node. When running C4.5 or OC1, default settings are used. C4.5 can generate both decision trees and decision rules. We chose to use generation of decision rules.

We expose ELEM2-CBR to five *classification* problems, each of which contains a target concept; an example in a problem either belongs to the target concept or does not belong. *Problem 1* 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

[5] By incremental learning we mean the previously tested examples in the testing set are used in case-based reasoning to solve problems represented by later test cases.

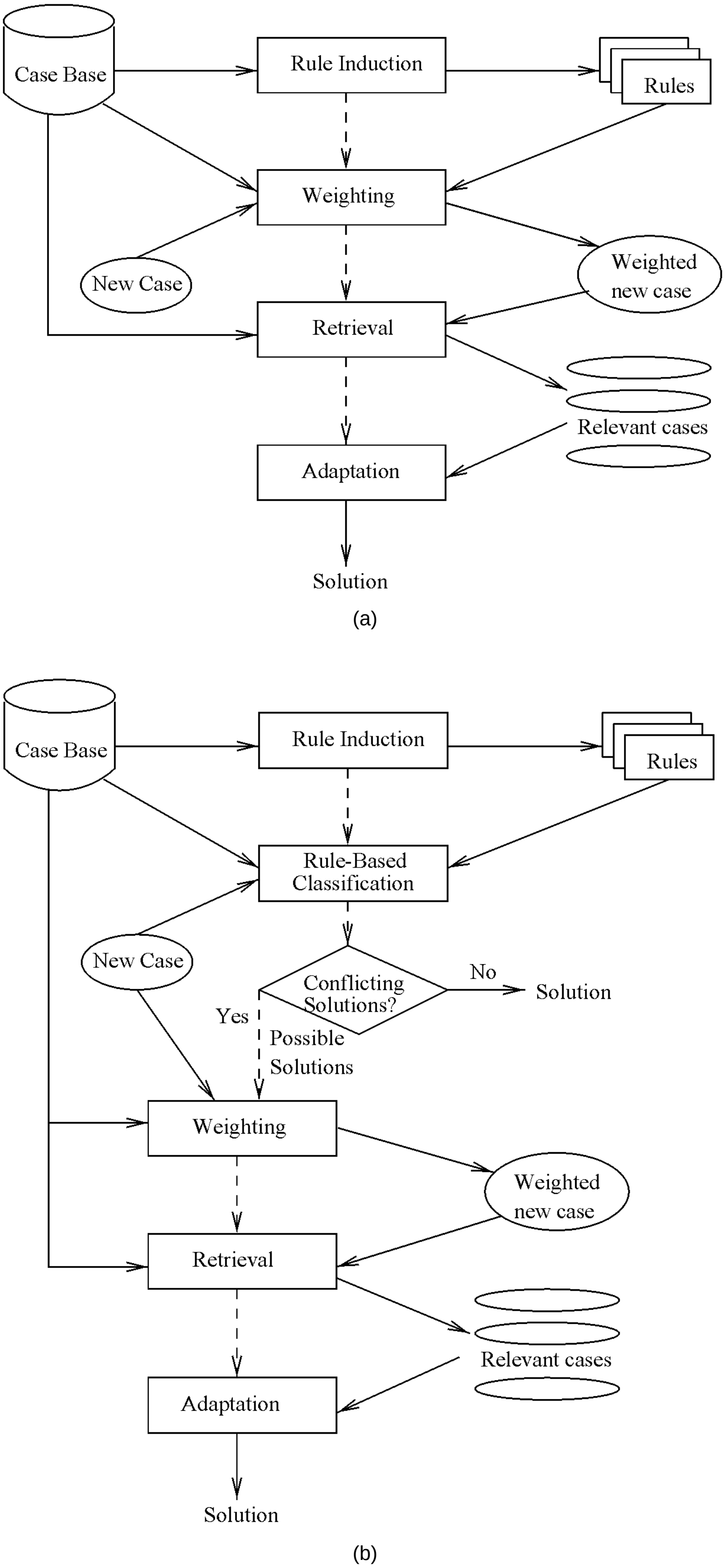

Fig. 1. ELEM2-CBR's numeric prediction (a) and classification (b).

to the concept." From the entire space of 1,024 possible examples, 256 were randomly chosen as training examples and the remaining as the testing set. *Problem 2* and *Problem 3* are designed to test ELEM2-CBR's ability to learn concepts with nonlinear boundaries. Each problem contains two continuous attributes representing two axes (*x* and *y*) in a

TABLE 1
PERFORMANCE COMPARISON ON DESIGNED CLASSIFICATION PROBLEMS

Problems	CBR-NW	CBR-PC	CBR-CC	ELEM2-CBR	C4.5	OC1
Problem 1	85.55%	64.84%	84.64%	**100%**	**100%**	85.16%
Problem 2	93.40%	70.40%	96.00%	98.80%	98.60%	**99.00%**
Problem 3	92.86%	74.06%	86.09%	**95.49%**	95.10%	89.47%
Problem 4	**96.24%**	83.46%	87.22%	95.86%	95.50%	95.86%
Problem 5	95.12%	76.32%	76.70%	**95.86%**	93.60%	88.35%

TABLE 2
PERFORMANCE COMPARISON ON REAL-WORLD CLASSIFICATION PROBLEMS

Datasets	CBR-NW	CBR-PC	CBR-CC	ELEM2-CBR	C4.5	OC1
australian	85.07%	**86.67**%	80.43%	85.65%	83.20%	83.33%
breast-cancer	**96.93**%	94.58%	96.78%	95.90%	95.00%	95.31%
glass	71.96%	52.34%	51.87%	**74.77**%	65.40%	70.56%
heart	81.48%	79.63%	80.37%	**82.22**%	78.90%	77.78%
tic-tac-toe	67.43%	65.34%	75.26%	99.37%	**99.50**%	n/a*
zoo	94.06%	58.42%	94.06%	**96.04**%	93.10%	89.11%
AVERAGE	82.82%	72.83%	79.80%	**88.99**%	85.85%	n/a

** The OC1 program is not applicable to the tic-tac-toe data set because the program takes only numeric valued attributes.*

two-dimensional space. An irrelevant attribute is added to each problem to test the algorithms' ability to tolerate irrelevant features. The target concepts of *Problem 2* and *Problem 3* 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 4* is the same as *Problem 3* except that there is no irrelevant feature in the data. *Problem 5* is derived from *Problem 4* by randomly adding 5 percent 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 test sets are shown in Table 1. The best result for each problem is highlighted in boldface. From Table 1, we note that ELEM2-CBR and C4.5 perform perfectly on *Problem 1*, while the three pure CBR algorithms do not perform well. This is because the concept in *Problem 1* has "rectangular" boundary regions and rule-induction algorithms are good at learning and representing these kinds of concepts, while pure CBR algorithms are not. Regarding the remaining four problems, ELEM2-CBR performs better than C4.5 which learns axis-parallel decision rules. This result is consistent with what we expected: Rules are not good at representing concepts with nonlinear boundaries. In *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. OC1 performs well on *Problems 2* and *4*, but not well on the other problems. We surmise that the oblique decision trees generated by OC1 do not well represent the axis-parallel categories in *Problem 1* and OC1's ability to handle noise is not as good as ELEM2-CBR and C4.5. In addition to artificial domains, we have also experimented with six real-world datasets from the UCI repository [19], for which the underlying concepts are unknown. Table 2 reports the results of *leave-one-out* evaluation on the six datasets.

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:

$$NP_1 : f(x, y, z) = x^2 + y^2 + z^2;$$

$$NP_2 : f(x, y) = \log_e(x) + \log_e(y); \text{ and}$$

$$NP_3 : f(x, y) = sin^{-1}(x) + cos^{-1}(y).$$

NP_4 is derived from NP_1 by randomly adding 5 percent prediction noise into the training set. For each problem, a set of examples are selected from the domain. One-third of the examples are randomly selected as training examples and the remaining examples selected as test samples. For each problem, the average of the relative errors made by each tested algorithm over the testing samples 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 also shown in Table 3.

5 CONCLUSIONS

We briefly summarized the salient features of rule-induction and case-based reasoning systems in the context of machine learning. We presented an extremely abbreviated survey of several of the more popular contemporary RI and CBR systems, including several which purport to be hybrid architectures of the two representations. We then presented ELEM2-CBR, a new method to integrate RI and CBR in which either method can be use to support the other; neither method need be in a sole support of supporting role.

TABLE 3
PERFORMANCE COMPARISON ON NUMERIC PROBLEMS

Problems	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%**
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%**

We explained our novel feature weighting function for case retrieval and our problem-solving procedures for both classification and numeric prediction. Our experiments have shown that ELEM2-CBR outperforms three other pure CBR systems, especially in learning concepts with hyper-rectangular boundaries or when the data set has irrelevant features or noisy training cases. In terms of learning non-hyper-rectangular regions, ELEM2-CBR also outperforms C4.5 and an oblique decision tree learning system, OC1.

Our experiments also indicate that ELEM2-CBR is doing much better than other methods in some of the datasets, such as “glass,” “heart,” and “machine,” but not in all. We surmise that the performance of ELEM2-CBR is related to the properties of the data set. We will investigate this relationship in the future, i.e., the relation between ELEM2-CBR and the nature of problems. Additional experiments with feature weighting and considerations of other novel learning strategies and methods to include in future hybrid architectures are planned.

ACKNOWLEDGMENTS

The authors are members of the Institute for Robotics and Intelligent Systems (IRIS) and wish to acknowledge the support of the Networks of Centers of Excellence Program of the government of Canada, the Natural Sciences and Engineering Research Council (NSERC), and the participation of PRECARN Associates Inc.

REFERENCES

[1] A. Aamodt, “Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches,” *Artificial Intelligence Comm.*, vol. 7, no. 1, 1994.

[2] K. Althoff, S. Wess, and R. Traphoner, “INRECA—A Seamless Integration of Induction and Case-Based Reasoning for Decision Support Tasks,” *Proc. Eighth Workshop German SIG on Machine Learning,* 1995.

[3] A. An, N. Cercone, C. Chan, N. Shan, and X. Huang, “ELEM: A Method for Inducing Rules from Examples,” *Proc. 15th Ann. Conf. British Computer Soc. Specialist Group on Expert Systems,* Cambridge, U.K., pp. 85-99, 1995.

[4] A. An, “Analysis Methodologies for Integrated and Enhanced Problem Solving,” PhD thesis, Dept. of Computer Science, Univ. of Regina, Sask., Canada, 1997.

[5] A. An and N. Cercone, “ELEM2: A Learning System for More Accurate Classifications,” *Proc. 12th Biennial Conf. Canadian Soc. Computational Studies of Intelligence, AI ’98,* Vancouver, B.C., Canada, 1998.

[6] E.R. Bareiss and C.C. Wier, “Protos: An Exemplar-Based Learning Apprentice,” *Proc. Fourth Int’l Workshop Machine Learning,* Irvine, Calif., 1987.

[7] M.M. Beaulieu, M. Gatford, X. Huang, S.E. Robertson, S. Walker, and P. Williams, “Okapi at TREC-5,” E.M. Voorhees, and D.K. Harman, eds., *Proc. Fifth Text REtrieval Conf. (TREC-5): Information Technology,* NIST Special Publication No. 500-238, 1997.

[8] W.S. Cooper, “On Selecting a Measure of Retrieval Effectiveness,” *J. Am. Soc. Information Science,* vol. 24, 1973.

[9] P. Clark, “Machine Learning: Techniques and Recent Developments,” A.R. Mirzai, ed., *Artificial Intelligence: Concepts and Applications in Eng.,* London, Chapman and Hall, 1990.

[10] P. Clark and R. Boswell, “Rule Induction with CN2: Some Recent Improvements,” *Proc. European Working Session Learning,* Porto, Portugal, pp. 151-163, 1991.

[11] R.H. Creecy, B.M. Masand, S.J. Smith, and D.L. Waltz, “Trading MIPS and Memory for Knowledge Engineering,” *Comm. ACM,* vol. 35, pp. 48-64, 1992.

[12] P. Domingos, “Rule Induction and Instance-Based Learning: A Unified Approach,” *Proc. IJCAI-95,* Montreal, Que., Canada, pp. 1,226-1,232, 1995.

[13] U.M. Fayyad and K.B. Irani, “Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning,” *Proc. IJCAI-93,* pp. 1,022-1,027, 1993.

[14] A.R. Golding and P.S. Rosenbloom, “Improving Rule-Based Systems Through Case-Based Reasoning,” *Proc. AAAI-91,* pp. 22-27, 1991.

[15] P. Koton, “Using Experience in Learning and Problem Solving,” PhD dissertation, Techncial Report No. MIT/LCS/TR-441, Laboratory of Computer Science, Massachusetts Inst. of Technology, 1989.

[16] P. Langley, ”Applications of Machine Learning and Rule Induction,” *Comm. ACM,* vol. 38, 1995.

[17] D.B. Leake, “CBR in Context: The Present and Future,” *Case-Based Reasoning: Experiences, Lessons, and Future Directions,* D.B. Leake, ed., Menlo Park, Calif., AAAI Press, 1996.

[18] R.S. Michalski, I. Mozetic, J. Hong, and N. Lavrac, “The Multi-Purpose Incremental Learning System AQ15 and Its Testing Application to Three Medical Domains,” *Proc. AAAI 1986,* pp. 1,041-1,045, 1986.

[19] P.M. Murphy and D.W. Aha, ”UCI Repository of Machine Learning Databases,” 1994; URL: http://www.ics.uci.edu/mlearn/MLRepository.html; for information, contact ml-repository@ics.uci.edu.

[20] S.K. Murthy, S. Kasif, and S. Salzberg, “A System for Induction of Oblique Decision Trees,” *J. Artificial Intelligence Research,* no. 2, pp. 1-32, 1994.

[21] J.R. Quinlan, *C4.5: Programs for Machine Learning,* Morgan Kaufmann, San Mateo, Calif., 1993.

[22] S.E. Robertson and K. Spark Jones, “Relevance Weighting of Search Terms,” *J. Am. Soc. Information Science,* vol. 27, pp. 129-146, 1976.

[23] S.E. Robertson, “The Probability Ranking Principle in IR,” *J. Documentation,* vol. 33, no. 4, 1977.

[24] S.E. Robertson, S. Walker, and M. Hancock-Beaulieu, “Large Test Collection, Experiments on an Operational, Interactive System: OKAPI at TREC,” *Information Processing and Management,* vol. 31, no. 3, pp. 345-360, 1995.

[25] D.B. Skalak and E.L. Rissland, “Inductive Learning in a Mixed Paradigm Setting,” *Proc. AAAI-90,* pp. 840-847, 1990.

[26] I. Watson, “Case-Based Reasoning Tools: An Overview,” *Proc. Second U.K. Workshop Case-Based Reasoning,* 1996.

[27] J. Zhang, “A Method that Combines Inductive Learning with Exemplar-Based Learning,” *Proc. Second Int’l IEEE Conf. Tools Artificial Intelligence,* Herndon, Va., pp. 31-37, 1990.

Nick Cercone received his BS degree in engineering science from the University of Steubenville (Ohio) in 1968, the MS degree in computer and information science from Ohio State University in 1970, and the PhD degree in computing science from the University of Alberta (Canada) in 1975. He is now a professor and chair in the Department of Computer Science at the University of Waterloo (Ontario, Canada), having recently completed his term as associate vice president (research), dean of graduate studies, and international liaison officer at the University of Regina (Saskatchewan, Canada). He formerly served as director of the Center for Systems Science at Simon Fraser University and chair of its School of Computer Science. His research interests include natural language processing, knowledge-based systems, knowledge discovery in databases, and design and human interfaces. He worked for IBM in 1969 and 1971 on design automation. He has authored 200 technical papers, is co-editor of *Computational Intelligence*, and serves on the editorial board of six journals. He is past president of the Canadian Society of Computational Studies of Intelligence (CSCSI/SCEIO), the Canadian Society for Fifth-Generation Research, and the Canadian Association for Computational Science (CACS/AIC). He served on the Canadian Genome Assessment and Technology Board, the CANARIE Board, CanWest, the institute for Robotics and Intelligent Systems (IRIS) Research Committee, the Saskatchewan Research Council Board, and the Regina Economic Development Authority (information technology). He now serves on committees of NSERC, CFI, CITO, and the National Science Foundation. In 1996, he won the Canadian Artificial Intelligence Society's Distinguished Service Award. He is a senior member of the IEEE, and a member of the ACM, IEEE Computer Society, AAAI, AISB, AGS, and ACL.

Aijun An received her BS and MS degrees in computer science at Xidian University in Xi'an, China, and the PhD degree in computer science from the University of Regina in Saskatchewan, Canada, in 1997. She joined the University of Waterloo (Canada) in 1997 and has since worked as a postdoctoral fellow in the Computer Science Department. Her research interests include machine learning, data mining, case-based reasoning, and knowledge engineering.

Christine Chan received MSc degrees in computer science and management information systems at the University of British Columbia (Canada) in 1986 and 1988, respectively. She received the PhD degree in the interdisciplinary studies of computer science, philosophy, and psychology from Simon Fraser University in 1992. In 1993, she joined the Department of Computer Science at the University of Regina (Saskatchewan, Canada), where she is currently an associate professor of computer science. She is also an adjunct scientist at the Telecommunications Research Laboratories (TRLabs). Her research interests include industrial applications of artificial intelligence, knowledge engineering, object-oriented methodologies for knowledge-based systems development, and educational instructional software.